TIETO- JA SÄHKÖTEKNIIKAN TIEDEKUNTA

Tuomas Holmberg

AKTIIVISEN OPPIMISEN NOPEUTTAMINEN DATAN VISUALISOINNILLA

Diplomityö
Tietotekniikan tutkinto-ohjelma
Huhtikuu 2020

Holmberg T. (2020) Aktiivisen oppimisen nopeuttaminen datan visualisoinnilla. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 59 s.

TIIVISTELMÄ

Massadatan käytön lisääntyminen, tietokoneiden laskentatehon kasvaminen ja koneoppimismenetelmien kehittyminen ovat johtaneet koneoppimissovellusten yleistymiseen. Nykyään koneoppimista käytetään monissa eri datan analysoimista ja ennustamista vaativissa tehtävissä esimerkiksi puheentunnistuksessa, luonnollisen kielen käsittelyssä, lääketieteellisessä kuvantamisessa, hakukoneissa ja konenäössä. Datasta oppimisen kannalta tärkeimpiä koneoppimisen osa-alueita ovat ohjattu, ohjaamaton ja puoliohjattu oppiminen.

Opetettavat koneoppimisen mallit vaativat yleensä valtavan määrän opetusdataa, jotta päästäisiin hyviin tuloksiin. Ohjatussa oppimisessa datan hankkimisen lisäksi jokainen yksittäinen datanäyte tarvitsee luokkatiedon, jonka ihminen joutuu antamaan käsin. Tämä työläs ja aikaavievä kategorisointiprosessi johtaa usein virheisiin, sillä ihmisen on vaikea pysyä johdonmukaisena rajatapausnäytteiden kanssa.

Aktiivisessa oppimisessa ihminen osallistuu oppimisprosessiin kategorisoimalla ainoastaan informatiivisimmat datanäytteet. Tässä datan visualisointi voi tukea opetuspäätöksissä ja parantaa tuloksia pienemmällä opetusdatan määrällä.

Datan visualisoimista varten kehitettiin Python-ohjelmistotyökalu, joka tuottaa kaksiulotteisia esityksiä. Tämä tuo luokitteluongelman ihmiselle helpommin ymmärrettäväksi samalla pienentäen erityisesti virheellisesti kategorisoitujen näytteiden lukumäärää.

Visualisoinneilla nopeutettiin mallin opettamista merkittävästi. MNIST-datalla opetetulla satunnaismetsäluokittelijalla päästiin 80 % luokittelutarkkuuteen 156 opetusnäytteellä ilman visualisointeja ja sama luokittelutarkkuus saavutettiin 134 opetusnäytteellä visualisointeja käyttämällä. Tosielämän epäbalansoidun oksadatan satunnaismetsäluokittelu 60 % Cohenin kappakertoimella vaati 127 näytettä ilman visualisointeja ja 63 näytettä visualisointien kanssa. Tutkimuksen yhteydessä kerätty madonmunanäytteistö luokittui satunnaismetsällä 90 % Cohenin kappakertoimella 95 opetusnäytteellä ilman visualisointeja, mutta visualisointityökalun avulla riitti 65 näytettä.

Tämä työ paljasti datan visualisoinnista saatavan edun erityisesti silloin, kun parhaat esitystavat eivät ole tunnettuja käytettävälle datajoukolle.

Avainsanat: tekoäly, puoliohjattu oppiminen, dimensionaalisuuden vähentäminen, monikerran oppiminen, esitystavan oppiminen

Holmberg T. (2020) Using Data Visualization to Accelerate Active Learning. University of Oulu, Degree Programme in Computer Science and Engineering, 59 p.

ABSTRACT

The increased use of big data, improved computational power of computers, and the development of machine learning techniques have led to the wider employment of machine learning based applications. For the time being machine learning is used for various data analysis and prediction tasks, for example, speech recognition, natural language processing, image analysis, and search engines. The machine learning approaches include supervised, unsupervised and semi-supervised schemes, each with different level of human effort.

Machine learning may require an enormous amount of data to achieve good results. In supervised learning, in addition to data acquisition, each data sample needs to be labeled, which has to be done manually by humans. This laborious and time-consuming annotation operation often leads to errors since for humans it is hard to stay logical while labeling borderline samples.

In active learning, humans participate in the learning process by labeling only the most informative data samples. Here data visualization can support in decisions improving results with less training data.

In the context of this thesis, an approach and a software tool for visualizing high dimensional data were developed. The solution brings the structure of data easier to understand, and improves the accuracy of labeling.

The approach was shown to speed up the training process of random forest classification in three comparable experiments. With MNIST hand written numerals data 80% accuracy was reached with 156 training samples without visualizations, while 134 training samples sufficed with the support of the developed tool. For an unbalanced wood material data set Cohen's kappa coefficient of 60% was reached with 127 training samples without visualizations, while only 63 were needed with visualizations. For worm egg data gathered during the research reaching Cohen kappa of 90% required only 65 samples using the visualization approach, but otherwise 95.

This work demonstrated the advantages of data visualization especially when the best representations for data are not known and are still explored.

Keywords: artificial intelligence, semi-supervised learning, dimensional reduction, manifold learning, representation learning

SISÄLLYSLUETTELO

ALKULAUSE

Tämä diplomityö on tehty Oulun yliopiston konenäön ja signaalianalyysin tutkimusyksikössä (CMVS) arvioimaan datan visualisoinneista saatavaa hyötyä aktiivisessa oppimisessa. Aluksi haluan kiittää ohjaajaani professori Olli Silvéniä mahdollisuudesta toimia hänen tutkimusryhmässään ja tuesta ja ohjauksesta diplomityön ja koko maisterivaiheen opintojeni aikana. Tutkimusryhmässä työskennellessäni olen saanut osallistua moniin tekoälyprojekteihin ja opetustyöhön ja sitä kautta kehittää osaamistani. Kiitos kuuluu myös professori Tapio Seppäselle diplomityöni tarkastamisesta. Kiitos Suomen Ruokaviraston tutkimusprofessori Antti Oksaselle, joka antoi luvan käyttää diplomityössäni Oulun tutkimusyksikössä kerättyä loiseläinten munien mikroskooppikuvista koostuvaa dataa. Lopuksi haluan kiittää perhettäni ja ystäviäni kaikesta tuesta opintojeni aikana.

Oulussa 2. huhtikuuta 2020

Tuomas Holmberg

LYHENTEIDEN JA MERKKIEN SELITYKSET

GRAY	Harmaasävykuva
HOG	Histogram of oriented gradients
k-NN	k-lähimmän naapurin luokittelija
LBP	Local binary pattern
LLE	Local linear embedding
LR	Logistinen regressio
MDS	Monidimensioskaalaus
PCA	Pääkomponenttianalyysi
RBF	Radial basis function kernel
RF	Satunnaismetsä
SOM	Itseorganisoiva kartta
t-SNE	t-distributed stochastic neighbor embedding
UMAP	Uniform manifold approximation and projection

κ	Cohenin kappakerroin
ϕ_{EL}	Epäluotettavuuden mitta
ϕ_{KL}	Kullback-Leibler divergenssi
$\phi_{KÄE}$	Kovan äänestyksen entropia
ϕ_{ME}	Maksimaalinen entropia
ϕ_{PM}	Pienimmän marginaalin mitta
$\phi_{PÄE}$	Pehmeän äänestyksen entropia

m_c	Itseorganisoivan kartan parhaiten sopiva painovektori
m_i	Itseorganisoivan kartan painovektori i
$p(y\|x)$	Mallin ennustamat luokkien posterioritodennäköisyydet
S	Hajontamatriisi
X^L	Kategorisoidut datanäytteet
X^U	Kategorisoimattomat datanäytteet
$X^{U'}$	Kategorisoimattomasta datasta valittu osajoukko

1. JOHDANTO

Useat koneoppimisella toteutettavat sovellukset edellyttävät suuren määrän dataa oppiakseen taustalla olevan ilmiön mallin. Isojen datamäärien keräämiseen liittyy monesti rajoitteita, jotka tulee ottaa huomioon. Datan tallentaminen tietokoneelle on helppoa mutta sen kategorisoiminen (engl. data annotation) vaatii lähes poikkeuksetta valtavasti työtä. Tosielämän tilanteissa data on yleensä erittäin kompleksista sisältäen toisiaan muistuttavia eri luokkien näytteitä, mikä lisää riskiä kategorisoida näytteitä väärin.

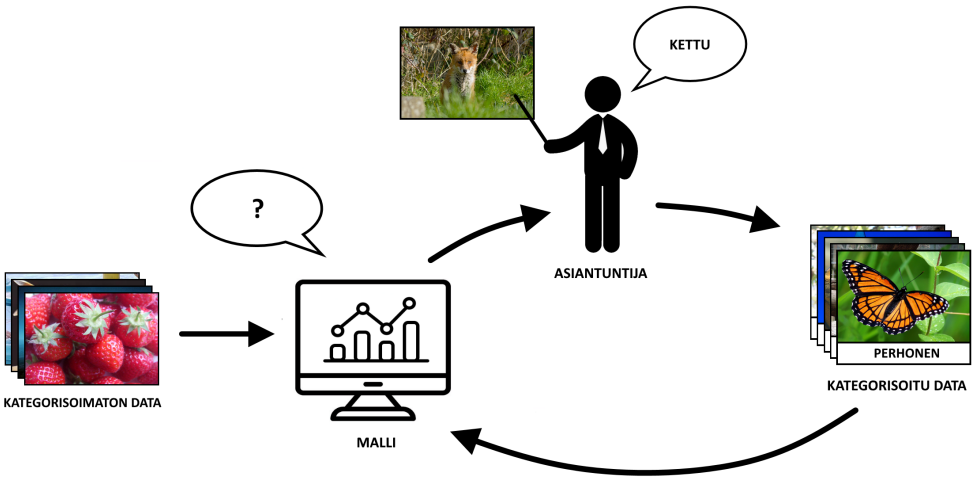
Esimerkiksi ImageNet-tietokannan alkuperäinen versio sisältää yli 14 miljoonaa internetistä kerättyä kuvaa yli 21 000 eri luokasta, kuten labradorinnoutajista, ambulansseista ja joulukoristeista [1]. Yksi luokka sisältää keskimäärin noin tuhat kuvaa. Kuvien koko, resoluutio, tausta ja objektien ulkonäkö, lukumäärä ja sijainti kuvassa vaihtelevat huomattavasti luokkien näytteiden välillä, mikä tekee luokitteluongelmasta haastavan. Kuvassa 1 on esimerkkikuvia ImageNet-tietokannan luokkien näytteistä.

Kuva 1. Esimerkkikuvia ImageNet-tietokannasta.

Koneoppiminen jaetaan yleisesti ohjattuun, ohjaamattomaan ja vahvistusoppimiseen. Ohjatussa oppimisessa malli opetetaan datalla, jolle on tarjolla datanäytteiden lisäksi lähtömuuttujat kuten luokkatiedot. Opetuksen jälkeen malli kykenee ennustamaan lähtömuuttujan tuntemattomalle datanäytteelle. Ohjaamattomassa oppimisessa malli opetetaan ilman lähtömuuttujia sisältävän datan perusteella, jolloin opetettu malli oppii datan rakenteisuuden. Vahvistusoppimisessa ei ole tarjolla dataa eikä lähtömuuttujia. Siinä toimintaympäristössään toimiva agentti saa reaaliaikaisesti positiivisia ja negatiivisia palautteita tekemistään liikkeistä. Ajan kuluessa agentti oppii yritysten ja erehdysten kautta löytämään optimaalisen strategian. Lisäksi tunnetaan ohjatun ja ohjaamattoman oppimisen välimaastoon sijoittuva puoliohjattu oppiminen. Siinä datan rakenteisuutta tai jakaumaa approksimoidaan ohjaamattoman oppimisen menetelmillä ja tätä tietoa hyödyntämällä päästään parempaan lopputulokseen ohjatun oppimisen luokittelijaa opetettaessa.

Puoliohjatun oppimisen erityistapaus, aktiivinen oppiminen valitsee koneoppimisen mallille kategorisoimattomien näytteiden datasta (engl. unlabeled data) vain ne näytteet, jotka ovat mallille vaikeimpia luokitella. Valitut näytteet syötetään sykleittäin asiantuntijalle (engl. oracle, expert, human annotator), joka kategorisoi vaikeimmat tapaukset kuvan 2 osoittamalla tavalla kasvattaen kategorisoitujen datanäytteiden lukumäärää (engl. labeled data). Malli oppii hahmottamaan luokkien väliset luokkarajat valitsemalla itse ne näytteet, jotka se haluaa oppia. Aktiivisen oppimisen toiminta perustuu oletukseen, että opetettu malli selviää helposti luokkien

tyypillisimpien näytteiden luokittelusta, joista data yleensä koostuu, tuntiessaan datasta valitut haasteellisimmat rajatapaukset.



Kuva 2. Aktiivisen oppimisen toimintaperiaate.

Keskittymistä vaativissa tehtävissä ihminen on altis monenlaisille virheille. Suuria tietokantoja kategorisoidessaan ihminen tekee helposti kirjoitus- ja näppäilyvirheitä. Tämän lisäksi ihminen tekee vaillinaiseen tietämykseen perustuvia systemaattisia virheitä, joissa jotain toista luokkaa muistuttavat näytteet ovat järjestelmällisesti kategorisoitu väärin. Virheiden korjaaminen vaatii aina kaikkien näytteiden uudelleen läpikäymistä.

Datan sisältäessä paljon väärin kategorisoituja näytteitä, ei voida olettaa, että millään ohjatun oppimisen menetelmällä saavutettaisiin huipputuloksia. Tähän liittyen tietotekniikassa käytetään käsitettä ”garbage in, garbage out”, jossa virheellinen data johtaa vääjäämättä virheelliseen lopputulokseen. Siksi on tärkeää, että käytetty data on mahdollisimman virheetöntä ja kuvaa tarkasti sen taustalla olevaa ilmiötä.

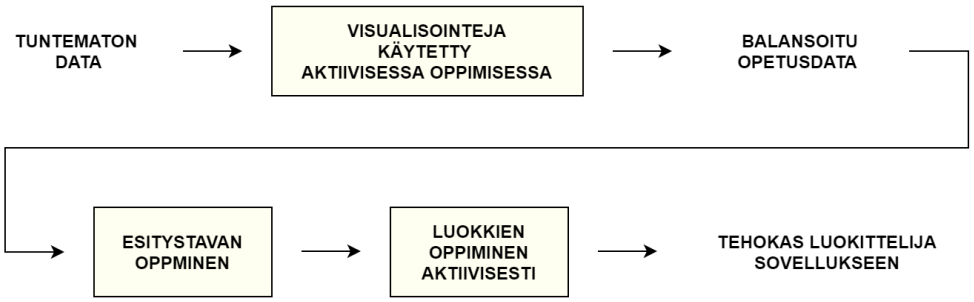
Jos luokkien välinen vaihtelevuus (engl. intra-class variability) on suurta ja luokkien sisäinen samankaltaisuus pientä (engl. inter-class similarity), on toisinaan alaan erikoistuneella asiantuntijallakin vaikeuksia pysyä johdonmukaisena rajatapausten kanssa. Kuvan 3 reunoilla on valittu ImageNet-tietokannasta luokkien saukko ja minkki tyyppiesimerkit ja keskellä on niiden rajatapauksia, joista on vaikea sanoa nopealla vilkaisulla, kumpaan luokkaan ne kuuluvat. Erityisesti aktiivisen oppimisprosessin alkuvaiheessa on vältettävä rajatapausten käyttöä, ettei mallin oppima tieto perustu pelkästään virhealttiisiin näytteisiin.

Kun datan tyypillisimmät ominaisuudet tai ulkonäkö vaihtelevat paljon eri ajanhetkinä tai eri ympäristöstä kerättynä, on datan uudelleen kerääminen ja luokkatietojen antaminen välttämättömiä toimenpiteitä hyvien luokittelutulosten saavuttamiseksi. Tämänkaltaisia tilanteita kohdataan esimerkiksi puuteollisuudessa, jossa sahatavaran laatu ja vikojen piirteet vaihtelevat huomattavasti vuodenajasta ja kasvuympäristöstä riippuen. Monissa tapauksissa kerättävän datan suuresta variaatiosta johtuen mallia opetettaessa on lähdettävä liikkeelle aina nollasta, eikä aiemmin kerättyä ja kategorisoitua dataa voida hyödyntää. Koko variaatiolla opetettaessa olisi riski, että luokittelutarkkuus heikkenisi.

Kuva 3. ImageNet-tietokannan luokkien minkki ja saukko-rajatapausnäytteiden kategorisointi tuottaa vaikeuksia.

Balansoidussa datassa jokaisessa luokassa on suurin piirtein yhtä monta näytettä. Jos luokittelijaa opetettaessa käytetään kovin epäbalansoitua dataa, saattaa luokittelija keskittyä yliedustetun luokan tunnistamiseen jättäen muut luokat vähemmälle huomiolle. Kun oppimisprosessi joudutaan aika ajoin aloittamaan alusta, on balansoidun datan takaaminen haasteellista. Jos ihminen osallistuisi kategorisoitavien näytteiden valintaan, hän voisi kontrolloida sekä datan laatua että balanssia. Näin tehtiin puumateriaalin luokkarajojen määrittämiseksi visuaalisiin havaintoihin perustuvassa menetelmässä itseorganisoivalle kartalle (SOM) [2].

Tässä diplomityössä on luotu ohjelmistotyökalu, jolla visualisoidaan ihmisen tekemien kategoriapäätösten vaikutusta mallin oppimiseen. Tämä tapahtuu projisoimalla korkeadimensioinen data kaksiulotteisessa kuvaajassa. Aiheen kannalta kiinnostavia tutkimuskysymyksiä ovat: 1) Mikä on minimäärä näytteitä, jolla päästään hyviin luokittelutuloksiin? 2) Kuinka paljon asiantuntija voi myötävaikuttaa mallin oppimiseen, kun tarjolla on visuaalinen kuvaus datasta? 3) Kuinka lähelle luokkarajaa asiantuntijan kategorisoimat näytteet sijoittuvat? Kuvassa 4 on havainnollistettu datan visualisoinneista saatavan hyödyn merkitys. Ennestään tuntemattomalle datalle haetaan aktiivisella oppimisella datan visualisointeja apuna käyttäen hyvälaatuinen ja balansoitu opetusdata. Tämän jälkeen opetusdatan pohjalta opitaan datan esitystapa ja luokat. Lopuksi opittua tehokasta luokittelijaa voidaan käyttää sovelluksena spesifiseen luokittelutehtävään.

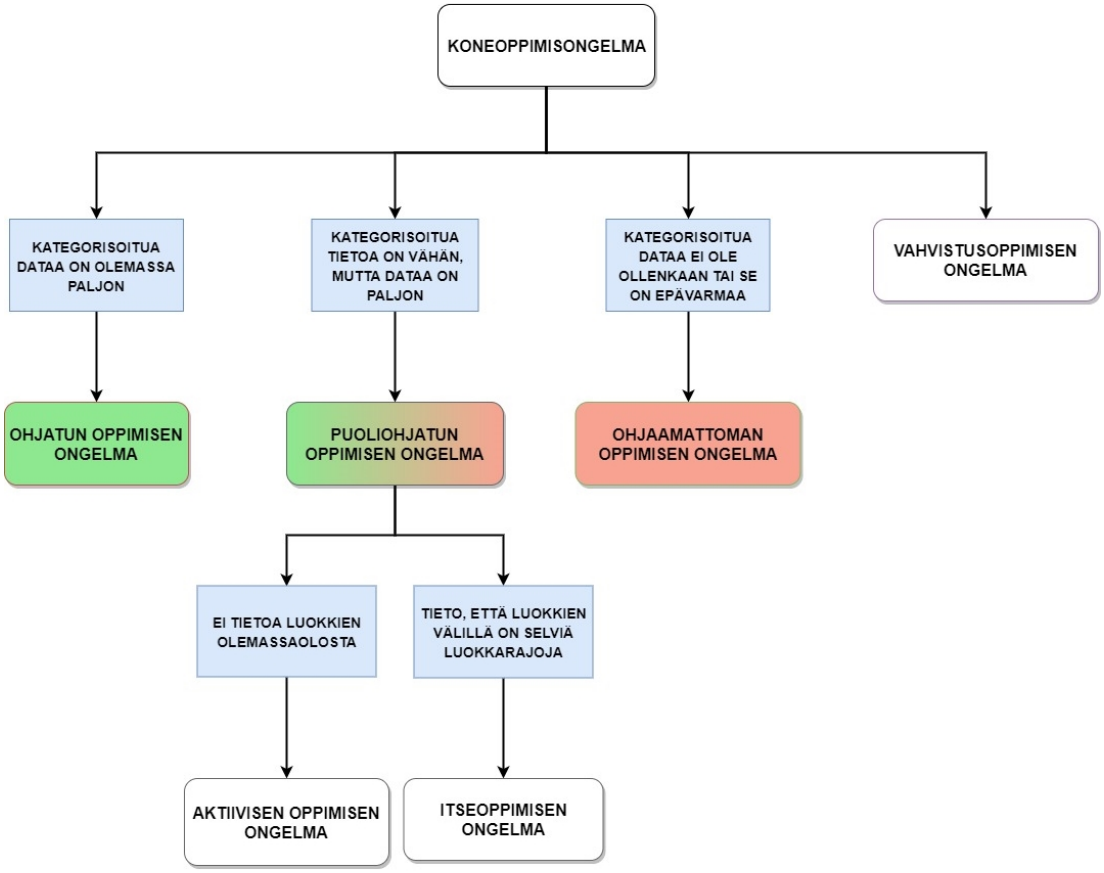


Kuva 4. Datan visualisoinneista saatavan hyödyn merkitys tehokasta luokittelijaa luotaessa.

Luvussa 2 ”Datalähtöinen koneoppiminen” esitellään koneoppimisen ja aktiivisen oppimisen menetelmiä datan näkökulmasta. Luvussa 3 ”Dimensionaalisuuden vähentämismenetelmät” käydään läpi eri dimensionaalisuuden vähentämismenetelmien toimintaperiaatteita ja arvioidaan lopuksi, mitkä niistä soveltuvat parhaiten aktiiviselle oppimiselle hyödyllisiin visualisointeihin. Luvussa 4 ”Datan visualisoiminen” esitellään diplomityössä käytetty data sekä sen analysoimiseen luotu datan visualisoimistyökalu. Luvussa 5 ”Tulokset ja niiden arviointi” käydään läpi perinteisellä aktiivisella oppimisella ja datan visualisoinneilla saatuja tuloksia ja vertaillaan niitä keskenään. Lisäksi pyritään löytämään vastaukset esitettyihin tutkimuskysymyksiin. Lopuksi luvussa 6 ”Pohdinta” arvioidaan työn onnistumista ja alan tulevaisuuden tutkimuskohteita ja luvussa 7 ”Yhteenveto” tehdään johtopäätökset.

2. DATALÄHTÖINEN KONEOPPIMINEN

Koneoppiminen on tekoälyn osa-alue, jossa opetettu malli tekee päätöksensä oppimansa opetusdatan tai palautteiden perusteella ilman, että sitä on ohjelmoitu erikseen päätöksenteossa. Koneoppimisongelmat jaetaan käytettävän datan perusteella ohjatun oppimisen, ohjaamattoman oppimisen, puoliohjatun oppimisen ja vahvistusoppimisen ongelmiin kuvan 5 mukaisesti. Datalähtöisen koneoppimisen näkökulmasta pääpaino on ohjatussa ja ohjaamattomassa oppimisessa sekä niiden väliin sijoittuvassa puoliohjatussa oppimisessa.



Kuva 5. Koneoppimisongelmien jako ohjatun, ohjaamattoman, puoliohjatun ja vahvistusoppimisen ongelmiin.

2.1. Ohjattu oppiminen

Kun kategorisoitua eli lähtömuuttujat sisältävää dataa on paljon, on kyseessä ohjatun oppimisen ongelma. Ohjattu oppiminen jaetaan edelleen regressioon ja luokitteluun. Regressioanalyysissä opetettu malli ennustaa testidatan näytteille jatkuva-arvoisia lähtömuuttujia. Luokittelussa mallin ennustamat testidatan lähtömuuttujat ovat diskreettejä eli näytteiden luokkia.

Malli tuntee opettamisen jälkeen opetusdatan tärkeimmät ominaisuudet. Oli kyseessä regressio-ongelma tai luokitteluongelma, mallin suorituskyvyn mittaamiseksi käytetään testidataa, joka on opetetulle mallille ennestään tuntematonta. Malli ennustaa testidatalle ennusteita ja ennusteiden oikeellisuutta arvioidaan erilaisia mittasuureita käyttämällä. Lisäksi usein käytetään validointidataa opetettavan mallin optimaalisimpien hyperparametrien arvojen selvittämiseksi.

Johdannossa esitettyä ImageNet-tietokantaa pidetään yhtenä luokittelumenetelmien suorituskykytestinä. Vuonna 2012 julkaistu neuronien-pudotuskerroksia (engl. dropout layers) sisältävä konvoluutioneuroverkko AlexNet onnistui luokittelemaan ImageNet-tietokannasta valitun testidatan 84,68 % luokittelutarkkuudella oikein [3]. Tätä pidettiin tuolloin sensaatiomaisena tuloksena, sillä aiemmilla perinteisillä koneoppimisen ja konenäön menetelmillä päästiin parhaillaan 74,2 % luokittelutarkkuuteen [4].

2.2. Ohjaamaton oppiminen

Ohjaamattomassa oppimisessa dataa ei ole kategorisoitu tai tieto näytteiden luokasta on epävarmaa. Ohjaamaton oppiminen jaetaan edelleen klusterointiin ja dimensionaalisuuden vähentämiseen. Klusteroinnissa algoritmi jakaa datan omiin ryhmiinsä perustuen täysin sen rakenteisuuteen. Tällöin samanlaiset eli data-avaruudessa lähellä toisiaan olevat näytteet klusteroidaan samaan ryhmään. Dimensionaalisuuden vähentämisessä pudotetaan korkeadimensioisten datanäytteiden piirteiden lukumäärää samalla menettäen mahdollisimman vähän merkittävää informaatiota. Usein ohjaamattoman oppimisen sovelluksissa käytetään ensin dimensionaalisuuden vähentämistä ja sen jälkeen klusteroidaan data matalammassa dimensiossa.

2.3. Puoliohjattu oppiminen

Puoliohjatussa oppimisessa liikutaan nimensä mukaisesti ohjatun ja ohjaamattoman oppimisen välimaastossa. Siinä mallin opettamisessa on tarjolla vain vähän kategorisoitua dataa X^L ja paljon kategorisoimatonta dataa X^U. Puoliohjatussa oppimisessa voidaan esimerkiksi selvittää karkeasti datan luokkarajat ohjatun oppimisen algoritmilla kategorisoitua dataa X^L käyttämällä ja hienosäätää niitä tarkemmin kategorisoimattoman datan X^U jakauman perusteella.

Puoli-ohjatun oppimisen mallit voidaan jakaa mallin luonteen perusteella generatiivisiin ja diskriminatiivisiin puoliohjatun oppimisen menetelmiin. Generatiiviset menetelmät mallintavat luokkien ehdollista todennäköisyyttä $p(x|y)$ ohjaamattoman oppimisen avulla [5]. Kun luokkien ehdolliset todennäköisyydet ovat selvillä, posterioritodennäköisyydet saadaan Bayesin teoreeman kautta.

$$p(y|x) = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dy}, \tag{1}$$

missä $p(y|x)$ on luokkien posterioritodennäköisyydet,
$p(x|y)$ on luokkien ehdolliset todennäköisyydet ja
$p(y)$ on luokkien prioritodennäköisyydet.

Diskriminatiiviset menetelmät eivät arvioi datan riippuvuutta luokasta, vaan ne keskittyvät arvioimaan suoraan posterioritodennäköisyyttä $p(y|x)$. Perinteisesti puoliohjattua oppimista on käytetty ehdollista todennäköisyyttä arvioivissa generatiivisissa malleissa, mutta nykyään se on yleistynyt myös diskriminatiivisten koneoppimismallien parissa. Hyviä tuloksia on saavutettu diskriminatiivisilla ydinmenetelmillä, kuten transduktiivisella tukivektorikoneella [6] tai Laplacen tukivektorikoneella [7]. Laskennallisesti kuitenkin tehokkaampi diskriminatiivinen menetelmä on puoliohjattu neuroverkko [8], jollaisella on onnistuttu käsittelemään jopa miljoonia kategorisoimattomia näytteitä siedettävässä ajassa.

Itseoppiminen (engl. self-training) on puoliohjatun oppimisen menetelmä, jossa kasvatetaan opetusdatan määrää luokittelemalla kategorisoimattomia näytteitä X^U ohjatusti opetetulla mallilla ja lisäämällä ne kategorisoitujen näytteiden joukkoon X^L. Joukkoon lisätään ainostaan ne näytteet, jotka omaavat suurimman varmuuden luokkatiedoistaan. [9]. Itseoppimisen käyttäminen edellyttää, että luokat ovat selvästi erotettavissa toisistaan. Sen toimintaperiaate on vastakkainen aktiivisen oppimisen epävarmuuteen perustuvalle näytteistykselle (engl. uncertainty sampling), jossa valitaan kategorisoitavaksi epävarmimmat näytteet. Asiaa käsitellään tarkemmin omassa kohdassa 2.4.4 ”Epävarmuuteen perustuva näytteistys”.

2.4. Aktiivinen oppiminen

Tässä osiossa on seurattu Burr Settlesin teoksen ”Active Learning” esitysjärjestystä [10], joka käsittelee kattavasti eri aktiivisen oppimisen menetelmiä. Perinteiselle aktiiviselle oppimiselle vaihtoehtoinen esitysratkaisu ja toisenlainen näkökulma on ns. kerta-ajo aktiivinen oppiminen (engl. single-pass active learning) [11], jossa asiantuntijalle tarjotaan yhdellä kertaa datan epävarmimmat näytteet.

Aktiivinen oppiminen tutkii oppimisjärjestelmien kehittymistä oppimisprosessin edetessä. Sen toiminnan pääolettamus on, että alkuperäisestä datasta oikein valitulla huomattavasti pienemmällä osajoukolla saavutetaan yhtä hyviä tai parempia luokittelutuloksia kuin opetettaessa malli koko datalla. Aktiivista oppimista käytetään usein tilanteissa, joissa kategorisoitua dataa X^L on tarjolla erittäin vähän ja kategorisoimatonta dataa X^U on saatavilla lisää reaaliaikaisesti. Tällöin viime aikoina laajasti käytössä olleilla syväoppimisen neuroverkkomalleilla ei päästä kategorisoidun datan vähyyden takia hyviin tuloksiin ja on käytettävä perinteisiä koneoppimisen algoritmeja. Poiketen itseoppimisesta aktiivisen oppimisen käyttäminen ei myöskään vaadi tietoa luokkien olemassaolosta.

Tosielämän datat sisältävät yleensä paljon toisiaan muistuttavia, helposti luokiteltavia näytteitä, jotka eivät tarjoa opetettavalle mallille lisäinformaatiota

ja vain vähän informatiivisia tapauksia. Onnistumalla löytämään datasta nämä rajatapaukset, voidaan syöttää ainoastaan ne asiantuntijan kategorisoitavaksi. Ihminen tekee yhteistyötä tietokoneella laskettavan mallin kanssa antamalla oppimiskiertoon kategoriapäätöksiä ja sitä kautta parantaa mallin toimintaa.

Algoritmissa 1 on esitetty aktiivisen oppimisen toimintaperiaate, joka vaihtelee hieman käytettävästä oppimisstrategiasta riippuen. Alkutilanteessa on valmiiksi kategorisoitu muutamia näytteitä joukkoon X^L ja joukko X^U sisältää paljon kategorisoimattomia näytteitä. Aktiivisen oppimisen prosessi noudattaa sykliä, jossa opetetaan koneoppimisen malli ϕ_t kategorisoiduilla näytteillä X^L. Malli generoi joko aivan uusia keinotekoisia näytteitä tai valitsee suuresta määrästä kategorisoimattomia näytteitä kaikista informatiivisimmat asiantuntijan kategorisoitavaksi. Näin ollen koneoppimisen malli valitsee tai tuottaa itse näytteet, jotka aiheuttavat sille eniten ongelmia ja kaipaa niiden tunnistuksessa asiantuntijan avustusta. Syklin lopuksi lisätään asiantuntijan kategorisoimat näytteet joukkoon X^L ja poistetaan ne joukosta X^U. Prosessi jatkuu niin kauan, kunnes malli saavuttaa halutun luokittelutarkkuuden tai alussa valittu syklien lukumäärä tulee täyteen.

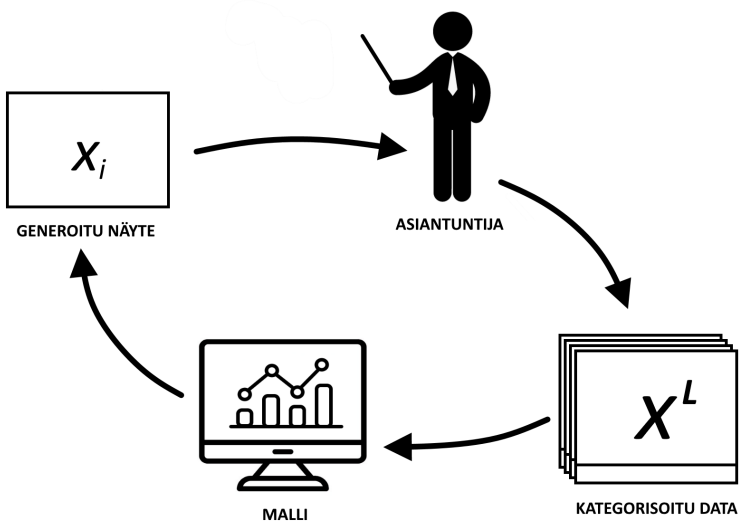
Algoritmi 1. Aktiivinen oppimisen toimintaperiaate

```
1 Alussa joukko X^L sisältää kategorisoituja näytteitä ja joukko X^U
    kategorisoimattomia näytteitä
2 for t = 1 to n do
3 |   Opetetaan malli φ_t kategorisoiduilla näytteille X^L
4 |   Malli valitsee epävarmimman näytteen joukosta X^U tai generoi uuden
  |     näytteen x_i
5 |   Valittu tai generoitu näyte x_i syötetään asiantuntijan kategorisoitavaksi
6 |   Kategorisoitu näyte lisätään joukkoon X^L ja poistetaan joukosta X^U
7 end
```

Aktiivinen oppiminen jaetaan kolmeen pääsuuntaukseen sen oppimisstrategian perusteella, joita ovat kyseltäväksi esitettävien näytteiden synteesi (engl. membership query synthesis), näytteiden virtaukseen perustuva näytteistys (stream-based selective sampling) ja näytejoukkoihin perustuva näytteistys (engl. pool-based selective sampling).

2.4.1. Kyseltäväksi esitettävien näytteiden synteesi

Kyseltäväksi esitettävien näytteiden synteesissä tunnetaan kategorisoitujen datanäytteiden X^L muodostama data-avaruus. Tästä generoidaan uusia keinotekoisia näytteitä x_i asiantuntijan kategorisoitavaksi kuvan 6 mukaisesti. Asiantuntijan kategorisoidessa generoituja datanäytteitä saadaan koneoppimisella määritettyä tarkemmin luokitteluongelman luokkarajat.



Kuva 6. Kyseltäväksi esitettävien näytteiden synteesi.

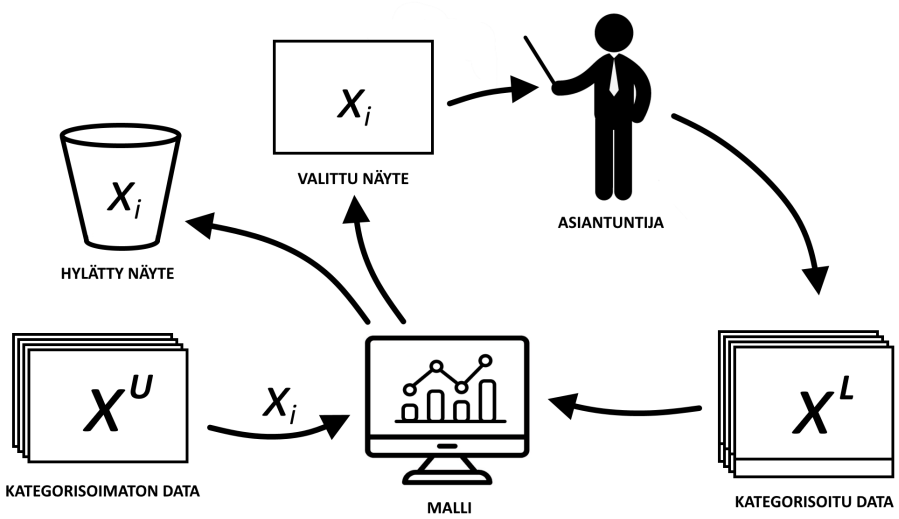
Informatiivisten näytteiden generoiminen siten, että ne ovat tunnistettavissa johonkin tiettyyn luokkaan, ei ole aina yksiselitteistä. Esimerkkinä tästä on muuntuvalla autoenkooderilla (engl. variational autoencoder) [12] tuotetut interpolaatiot numerosta toiseen käsinkirjoitettujen numeroiden MNIST-datasta [13]. Kuten kuvasta 7 nähdään, kaikkia keinotekoisesti interpoloituja näytteitä ei pystytä kategorisoimaan mihinkään aiemmin tunnettuun luokkaan. Esimerkiksi interpoloiduista käsinkirjoitettujen numeroiden 6 ja 7 välisistä näytteistä on vaikea sanoa, muistuttavatko keskimmäiset interpolaatiot numeroa 6, 7 vai 0. Tästä herääkin kysymys: tuovatko näytteet, joita edes asiantuntija ei osaa luokitella varmuudella oikein, lisää informaatiota opetettavalle mallille vai johtavatko ne sitä harhaan?

Kuva 7. Esimerkkinäytteitä käsinkirjoitettujen numeroiden 6, 7, 8 ja 9 synteesistä muuntuvalla autoenkooderilla.

2.4.2. Näytteiden virtaukseen perustuva näytteistys

Näytteiden virtaukseen perustuvassa näytteistyksessä oletetaan, että näytteiden kategorisoiminen on erityisen kallisarvoinen operaatio ja uusien kategorisoimattomien näytteiden hankkiminen on käytännössä katsoen ilmaista. Tähän oletukseen

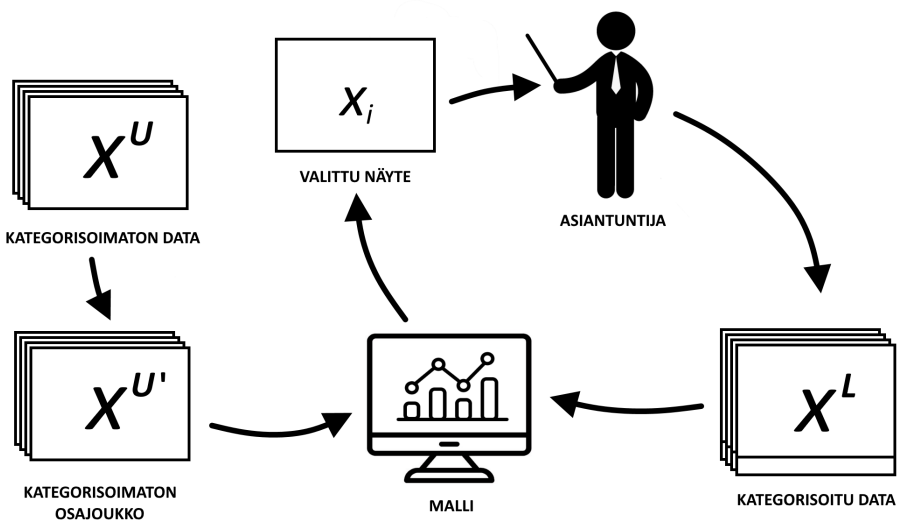
perustuen arvioidaan kategorisoimattomasta datasta X^U näyte kerrallaan, syötetäänkö näyte asiantuntijan kategorisoitavaksi vai hylätäänkö se kokonaan kuvan 8 mukaisesti. Päätöksessä näytteen valitsemiseksi tai hylkäämiseksi voidaan käyttää esimerkiksi epävarmuuteen perustuvaa näytteistystä tai komiteakyselyä (engl. query-by-committee), joissa molemmissa tapauksissa hyödynnetään kategorisoitua dataa X^L valinnassa. Lisäksi menetelmä vaatii käytettävälle informaation mitalle kynnysarvon, jonka perusteella kategorisoimaton näyte joko hylätään tai syötetään kyseltäväksi. Väärin asetettu kynnysarvo johtaa joko liian monen näytteen hyväksymiseen, jolloin asiantuntijaa kuormitetaan turhaan tai liian monen näytteen hylkäämiseen, jolloin asiantuntijan koko potentiaalinen tietotaito jää hyödyntämättä.



Kuva 8. Näytteiden virtaukseen perustuva näytteistys.

2.4.3. Näytejoukkoihin perustuva näytteistys

Näytejoukkoihin perustuvassa näytteistyksessä valitaan kategorisoimattomien näytteiden datasta X^U pienempi osajoukko $X^{U'}$, josta syötetään kaikista suurimman informaation omaava näyte asiantuntijan kategorisoitavaksi kuvan 9 tapaan. Keskeisimpänä erona näytteiden virtaukseen perustuvaan näytteistykseen on, että nyt päätöstä näytteen syöttämiseksi asiantuntijalle ei tehdä jokaiselle näytteelle erikseen, vaan arvioidaan osajoukon kaikki näytteet ja valitaan niistä paras. Tämä menetelmä ei siis vaadi kynnysarvon säätämistä, vaikkakin osajoukon koolla on vaikutusta lopputulokseen. Lisäksi on otettava huomioon, että osajoukon koon ollessa liian iso, kuluu osajoukon näytteiden arvioimisessa suhteettoman kauan aikaa.



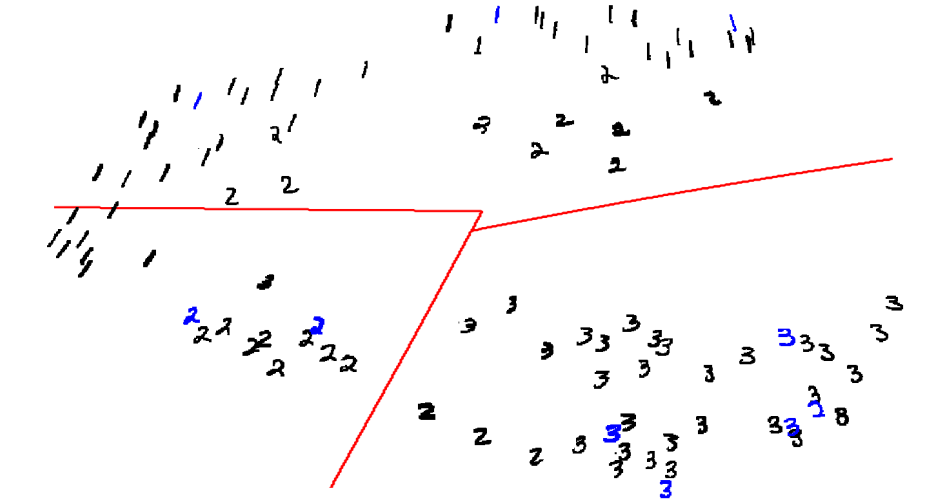
Kuva 9. Näytejoukkoihin perustuva näytteistys.

2.4.4. Epävarmuuteen perustuva näytteistys

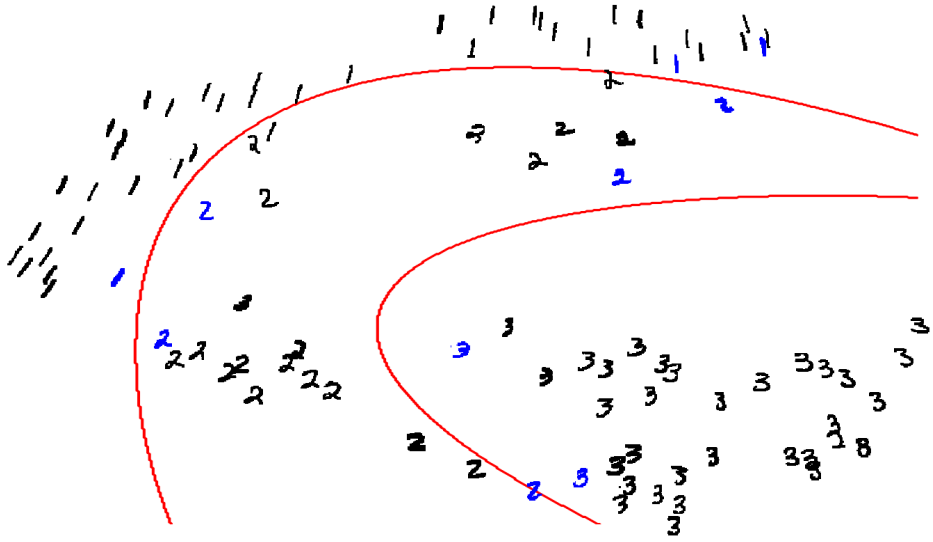
Epävarmuuteen perustuvassa näytteistyksessä mitataan, kuinka varma aktiivisesti opetettu malli on siitä, että kategorisoimaton näyte kuuluu johonkin aiemmin tunnettuun luokkaan. Valitsemalla opetusdataksi kaikista epävarmimmat eli luokkarajaa lähimpänä olevat näytteet, saadaan hahmoteltua luokkarajat tarkemmin kuin satunnaisella näytteistyksellä eli passiivisella oppimisella.

Kuvassa 10 käsinkirjoitettujen numeroiden MNIST-datasta on poimittu yhteensä 100 näytettä luokista 1, 2 ja 3 ja näytteet on projisoitu UMAP-algoritmilla kaksiulotteiseen avaruuteen. Opetusdatan 10 näytettä on esitetty sinisinä ja testidatan 90 näytettä mustina kirjoitusmerkkeinä. Ohjatun oppimisen algoritmin RBF-tukivektorikoneluokittelijan luomia luokkarajoja on havainnollistettu punaisilla käyrillä.

Kuvassa 10 (a) on käytetty satunnaista näytteistystä ja kuvassa 10 (b) epävarmuuteen perustuvaa näytteistystä opetusdatan näytteiden valinnassa. Kuten kuvista havaitaan, satunnaisella näytteistyksellä ei onnistuta mallintamaan tarkasti yksinkertaisen luokitteluongelman luokkarajoja ja luokittelutarkkuudeksi saadaan ainoastaan 75,6 %. Kun näytteet on valittu epävarmuuteen perustuvalla näytteistyksellä lähempää luokitteluongelman luokkarajoja, nousee luokittelutarkkuus arvoon 96,7 %.



(a)



(b)

Kuva 10. MNIST-datan luokkien 1, 2 ja 3 luokittelutulos kaksiulotteisessa avaruudessa, kun opetusdatan valitsemiseen käytetään (a) satunnaista näytteistystä ja (b) epävarmuuteen perustuvaa näytteistystä.

Binäärisessä kahden luokan luokitteluongelmassa riittää, että epävarmimmaksi näytteeksi valitaan se, jonka positiiviseksi (tai negatiiviseksi) ennustetun näytteen posterioritodennäköisyys on lähimpänä arvoa 0,5. Kolmen tai useamman luokan luokitteluongelmassa ei tämä näytteistysstrategia kuitenkaan enää sellaisenaan toimi. Seuraavaksi esitellään kolme yleisintä epävarmuuteen perustuvaa näytteistysstrategiaa.

Yksinkertaisin menetelmä on valita asiantuntijan kategorisoitavaksi kaikista suurimman *epäluotettavuuden* (engl. least confident) mitan saanut näyte [10]. Se lasketaan yksittäiselle datanäytteelle yhtälöllä (2) vähentämällä todennäköisyydestä 1,0 posterioritodennäköisyyden arvo ennusteen kuulumisesta todennäköisimpään

luokkaan. Lopuksi valitaan osajoukosta suurimman epäluotettavuuden mitan saanut näyte.

$$\phi_{EL} = \arg\max_{x} \ (1{,}0 - p(\hat{y_1}|x)), \tag{2}$$

missä ϕ_{EL} on epäluotettavuuden mitta ja
$\hat{y_1}$ on mallin ennustama todennäköisin luokka näytteelle x.

Epäluotettavuuden mitta ottaa huomioon vain näytteen ennusteen todennäköisimmän luokan posterioritodennäköisyyden, mutta ei huomioi ennusteen muiden luokkien todennäköisyyksiä. *Pienimmän marginaalin* (engl. smallest margin) mitta [14] ottaa huomioon kahden todennäköisimmän ennustetun luokan posterioritodennäköisyydet laskemalla niiden erotuksen yhtälön (3) mukaisesti. Osajoukosta valitaan lopuksi lähimpänä nollaa pienimmän marginaalin saanut näyte.

$$\phi_{PM} = \arg\min_{x} \ (p(\hat{y_1}|x) - p(\hat{y_2}|x)), \tag{3}$$

missä ϕ_{PM} on pienimmän marginaalin mitta ja
$\hat{y_1}$ ja $\hat{y_2}$ ovat mallin ennustamat näytteen x kaksi todennäköisintä luokkaa.

Jos luokkien näytteet on mahdollista sekoittaa kolmen tai useamman luokan kesken, ei pienin marginaali ole optimaalisin suure mittaamaan epävarmuutta. *Maksimaalisen entropian* (engl. maximum entropy) näytteistyksessä [15] huomioidaan ennusteen kaikkien luokkien ehdolliset todennäköisyydet yhtälön (4) mukaisesti. Yleisesti ottaen entropia on mitta, joka kuvaa epäjärjestyksen määrää. Mitä suurempi entropian arvo on, sen epävarmempi malli on näytteen luokasta. Osajoukosta valitaan lopuksi suurimman maksimaalisen entropian mitan saanut näyte.

$$\phi_{ME} = \arg\max_{x} \ (-\sum_{i=1}^{N} p(\hat{y_i}|x) \log p(\hat{y_i}|x)), \tag{4}$$

missä ϕ_{ME} on maksimaalinen entropia ja
$\hat{y_i}$ on mallin ennuste luokaksi y_i näytteelle x ja N on luokkien lukumäärä.

Tarkastellaan esimerkkitilannetta, jossa kolme luokkaa a, b ja c sisältävästä datasta on erotettu prosessin alussa viiden näytteen testidata. Koneoppimisen mallilla ennustetaan luokille posterioritodennäköisyydet ja lasketaan arvot epävarmuuteen perustuvilla näytteistysstrategioilla taulukkoon 1. Taulukossa on alleviivattu näyte, joka mitan perusteella lopuksi valitaan epävarmimmaksi näytteeksi. Pienimmän marginaalin mitan perusteella epävarmimmaksi näytteeksi valikoituu näyte 5. Epäluotettavuuden mitan ja maksimaalisen entropian käyttäminen johtavat näytteen 2 valintaan.

Taulukko 1. Esimerkki viiden näytteen testidatan luokkien a, b ja c ennustetuista todennäköisyyksistä ja lasketuista epävarmuuden mittojen arvoista

	$p(\hat{a}\|x)$	$p(\hat{b}\|x)$	$p(\hat{c}\|x)$	ϕ_{EL}	ϕ_{PM}	ϕ_{ME}
näyte 1	0,55	0,42	0,03	0,45	0,13	0,347
näyte 2	0,30	0,36	0,34	0,64	0,02	0,476
näyte 3	0,28	0,42	0,30	0,58	0,12	0,470
näyte 4	0,90	0,02	0,08	0,10	0,82	0,163
näyte 5	0,45	0,10	0,45	0,55	0,00	0,412

2.4.5. Komiteakysely

Komiteakyselyssä (engl. query-by-committee) jokaisen aktiivisen oppimisen syklin aikana opetetaan yhden koneoppimisen mallin sijaan joukko malleja eli komitea. Mallien opettamisen jälkeen ne ennustavat kategorisoimattomille datanäytteille X^U luokat tai posterioritodennäköisyydet kuulua eri luokkiin. Asiantuntijalle syötettäviksi näytteiksi valitaan ne, joista mallien komitealla on kaikista eriävin näkemys ennustettavan näytteen luokasta.

Mallien erimielisyyden mittaamiseksi voidaan käyttää kahdenlaista strategiaa: *Äänestyksen entropiaa* (engl. vote entropy) tai *keskiarvoistettua Kullback-Leibler divergenssiä*. Äänestyksen entropia noudattaa aiemmin esitettyä entropian yhtälöä (4) sillä erolla, että nyt eri mallien ennustamat posterioritodennäköisyydet datanäytteen luokaksi $\hat{y}_i$ lasketaan yhteen ja jaetaan komitean mallien lukumäärällä. Tätä menetelmää kutsutaan *pehmeän äänestyksen entropiaksi*, joka lasketaan yhtälöillä (5) ja (6). Mikäli posterioritodennäköisyyksiä ei ole saatavilla, voidaan niiden sijaan käyttää ennustettujen luokkien lukumäärää jaettuna komitean koolla. *Kovan äänestyksen entropia* lasketaan yhtälöllä (7).

$$p_{\mathcal{K}}(\hat{y}_i|x) = \frac{1}{K}\sum_{k=1}^{K} p_k(\hat{y}_i|x), \tag{5}$$

missä $p_{\mathcal{K}}(\hat{y}_i|x)$ on komitean mallien ennustama keskiarvoistettu posterioritodennäköisyys näytteen x luokaksi y_i ja
K on komitean mallien lukumäärä.

$$\phi_{PÄE} = \arg\max_x \ (-\sum_{i=1}^{N} p_{\mathcal{K}}(\hat{y}_i|x)\log p_{\mathcal{K}}(\hat{y}_i|x)), \tag{6}$$

missä $\phi_{PÄE}$ on pehmeän äänestyksen entropia ja
N on luokkien lukumäärä.

$$\phi_{K\ddot{A}E} = \arg\max_{x} \ (-\sum_{i=1}^{N} \frac{\ddot{A}(\hat{y}_i, x)}{K} \log \frac{\ddot{A}(\hat{y}_i, x)}{K}), \tag{7}$$

missä $\phi_{K\ddot{A}E}$ on kovan äänestyksen entropia ja
$\ddot{A}(\hat{y}_i, x)$ on mallien antamien äänten lukumäärä näytteen x luokaksi y_i.

Toinen strategia äänestystuloksen erimielisyyden mittaamiseksi on keskiarvoistettu Kullback-Leibler divergenssi, joka mittaa kahden eri todennäköisyysjakauman eroavaisuutta. Nyt yksittäisen mallin ennusteen posterioritodennäköisyyksiä luokaksi y_i verrataan komitean kaikkien mallien ennusteiden keskiarvoistettuun posterioritodennäköisyyteen. Keskiarvoistettu Kullback-Leibler divergenssi lasketaan yhtälöllä (8).

$$\phi_{KL} = \arg\max_{x} \ (\frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N} p_k(\hat{y}_i|x) \log \frac{p_k(\hat{y}_i|x)}{p_{\mathcal{K}}(\hat{y}_i|x)}), \tag{8}$$

missä ϕ_{KL} on keskiarvoistettu Kullback-Leibler divergenssi.

Tarkastellaan esimerkkinä kolmen luokan a, b ja c luokitteluongelmaa, jossa lasketaan taulukossa 2 kolmelle testinäytteelle pehmeän äänestyksen entropia, kovan äänestyksen entropia ja keskiarvoistettu Kullback-Leibler divergenssi kolmen mallin komitean posterioritodennäköisyyksistä. Kovan ja pehmeän äänestyksen entropian perusteella komitea valitsisi asiantuntijalle syötettäväksi näytteen 3. Keskiarvoistettua Kullback-Leibler divergenssiä käyttämällä vaikeimmin luokiteltava näyte komitean mukaan on näyte 1.

Taulukko 2. Esimerkki kolmen näytteen luokkien a, b ja c kolmella eri mallilla ennustetuista posterioritodennäköisyyksistä ja komitean eriävää mielipidettä kuvaavien mittojen arvojen laskemisesta

	k	$p_k(\hat{a}\|x)$	$p_k(\hat{b}\|x)$	$p_k(\hat{c}\|x)$	$\phi_{P\ddot{A}E}$	$\phi_{K\ddot{A}E}$	ϕ_{KL}
näyte 1	1	0,41	0,28	0,31	0,468	0,276	0,0045
	2	0,35	0,29	0,36			
	3	0,52	0,21	0,27			
näyte 2	1	0,04	0,05	0,91	0,217	0,000	0,0042
	2	0,10	0,08	0,82			
	3	0,11	0,04	0,85			
näyte 3	1	0,37	0,30	0,33	0,477	0,477	0,0022
	2	0,31	0,38	0,31			
	3	0,29	0,32	0,39			

2.4.6. Aktiivisen oppimisen mallit

On hyödyllistä vertailla koneoppimisessa käytettyjä luokittelijoita aktiivisen oppimisen näkökulmasta. 1990-luvun puolivälissä julkaistu tukivektorikone on yleisesti käytetty luokittelija sen hyvän suorituskyvyn ansiosta [16]. Kompleksisten luokitteluongelmien epälineaarisia luokkarajoja mallinnetaan ydinmenetelmällä, jossa alkuperäisen data-avaruuden näytteet kuvataan ydinfunktiolla korkeadimensioiseen avaruuteen. Suorittamalla tukivektorien laskenta korkeassa dimensiossa ja muuntamalla ne takaisin matalaan dimensioon, muuntuvat lineaariset vektorit epälineaarisiksi funktioiksi. Ydinfunktioina käytetään mm. polynomifunktiota, RBF:ää (engl. radial based function) ja sigmoid-funktiota. Tukivektorikoneen laskennallinen kompleksisuus $\mathcal{O}(\max(n,d) \cdot \min(n,d)^2)$ suurenee opetusnäytteiden n ja piirteiden lukumäärän d kasvaessa [17].

Eräs yksinkertaisimmista luokittelijoista on k-lähimmän naapurin menetelmä. Siinä luokkatiedottomalle testinäytteelle valitaan luokaksi sen k-lähimmän naapurin enemmistöluokka. Naapureiden etäisyyttä toisistaan arvioidaan usein euklidisen etäisyyden perusteella data-avaruudessa. Euklidinen k-lähimmän naapurin etäisyysmitta kärsii dimensionaalisuuden kirouksesta, jolloin samanlaisetkin datanäytteet monidimensioisessa piirreavaruudessa saattavat sijaita kaukana toisistaan. Siksi piirteiden muuntaminen mataladimensioisiksi piirrevektoreiksi on yleisesti käytetty toimenpide k-lähimmän naapurin luokittelijan luokittelutarkkuuden parantamiseksi. Luokittelun laskennallinen kompleksisuus mallin opettamiseksi on $\mathcal{O}(ndk)$, missä k on lähimpien naapureiden lukumäärä.

Satunnaismetsä on päätöspuista koostuva yhdistelmäluokittelija. Jokainen yksittäinen päätöspuu on opetettu satunnaisesti opetusdatasta näytteistämällä, joka menetelmänä tunnetaan nimellä bagging. Tämän lisäksi päätöspuut valitsevat satunnaisesti, mitä piirteitä ne käyttävät. Jotta erilaisia datan piirteiden ja näytteiden kombinaatioita saadaan generoitua riittävästi, tarvitaan paljon dataa. Lopuksi satunnaisesti generoidut päätöspuut ennustavat testinäytteelle luokan ja eniten ääniä saanut luokka valitaan ennusteeksi. Satunnaismetsän laskennallinen kompleksisuus uuden mallin opettamiseksi on $\mathcal{O}(md'nlog(n))$, missä m on generoitujen päätöspuiden lukumäärä ja d' on valittujen piirteiden lukumäärä yhdelle päätöspuulle.

Logistinen regressio on regressioanalyysin erityistapaus, jota käytetään yleensä nimestään huolimatta datan luokittelussa. Se pyrkii ennustamaan binääriselle luokitteluongelmalle sigmoid-funktiolla ja monen luokan luokitteluongelmalle softmax-funktiolla, millä todennäköisyydellä määritelty tapahtuma tulee tapahtumaan. Stokastisella gradientin vähentämismenetelmällä opetettaessa logistisen regression laskennallinen kompleksisuus on $\mathcal{O}(ndce)$, missä c on luokkien lukumäärä ja e on ajettujen jaksojen lukumäärä.

Kaikki neljä esitettyä luokittelumenetelmää vaativat riittävän määrän opetusnäytteitä, jotta opetusdatan ylioppimiselta vältytään. Niiden laskennallinen kompleksisuus on kuitenkin merkittävästi pienempi kuin ns. syväoppimismallien, mikä on aktiivisen oppimisen kannalta tärkein ominaisuus.

3. DIMENSIONAALISUUDEN VÄHENTÄMISMENETELMÄT

Kerättäessä dataa kiinnostavasta ilmiöstä, voi muuttujia kertyä tuhansia tai jopa miljoonia. Esimerkiksi kameran ottamassa 1000 x 1000 kokoisessa RGB-kuvassa on yhteensä 3 miljoonaa pikseliä eli 3 miljoonaa muuttujaa. Oppiakseen tunnistamaan tehokkaasti monidimensioisten datanäytteiden takana olevan ilmiön, on ylimääräisestä informaatiosta päästävä eroon pudottamalla muuttujien lukumäärää.

Dimensionaalisuuden vähentäminen on häviöllinen ohjaamattoman oppimisen osa-alue. Siinä pyritään pienentämään korkeadimensioisen datan muuttujien lukumäärää olennaista informaatiota menettämättä. Muunnoksen jälkeen datan tärkeimmät ominaisuudet voidaan selittää pienemmällä määrällä alkuperäisistä muuttujista valituilla tai muunnetuilla piirteillä. Kyseessä on siis datan tehokkaan esitystavan etsiminen.

Dimensionaalisuuden vähentämisessä on kaksi erilaista lähestymistapaa: piirteiden valitseminen (engl. feature selection) ja piirteiden irrotus (engl. feature extraction). Piirteiden valitsemisessa keskitytään löytämään alkuperäisistä muuttujista ne, jotka mallintavat mahdollisimman hyvin koko datan ominaisuuksia. Piirteiden irrotuksessa muunnetaan korkeadimensioinen data matalampidimensioiseksi yhdistämällä toistensa kanssa korreloivia muuttujia uusiksi piirteiksi.

Eri dimensionaalisuuden vähentämismenetelmät pyrkivät säilyttämään datan eri ominaisuuksia, sillä kaikkia niitä ei voida pitää. Ne voivat esimerkiksi pyrkiä minimoimaan keskineliövirhettä, säilyttämään datanäytteiden välisiä euklidisia etäisyyksiä tai ottamaan huomioon datanäytteiden naapuruusinformaation. Mikään yksittäinen dimensionaalisuuden vähentämismenetelmä ei ole muita menetelmiä selvästi parempi, vaan datasta riippuen ne toimivat eri tavalla. Aiemmat seikat huomioon ottaen, on siis lähes mahdotonta sanoa etukäteen testaamatta, mitä menetelmää tulisi käyttää.

Tarkastellaan seuraavaksi tarkemmin, miten eri dimensionaalisuuden vähentämismenetelmät suorittavat muunnoksen.

3.1. Pääkomponenttianalyysi

Pääkomponenttianalyysi (engl. principal components analysis, PCA) on tunnetuin lineaarinen dimensionaalisuuden vähentämismenetelmä, josta käytetään myös nimityksiä Hotelling-muunnos tai Karhunen-Loéve-muunnos [18]. Siinä korkeaulotteiseen data-avaruuteen määritellään uudet muuttujat eli ns. pääkomponentit, jotka ovat riippumattomia lineaarikombinaatioita kaikista datan muuttujista. Pääkomponentit valitaan siten, että ne kuvaavat suurimman osan datan vaihtelusta ja ovat ortogonaalisia toisiinsa nähden. Ensimmäisen pääkomponentin tulee selittää mahdollisimman suuren osan alkuperäisen datan vaihtelusta ja siitä seuraavat pääkomponentit selittävät mahdollisimman hyvin jäljelle jäävän datan vaihtelun.

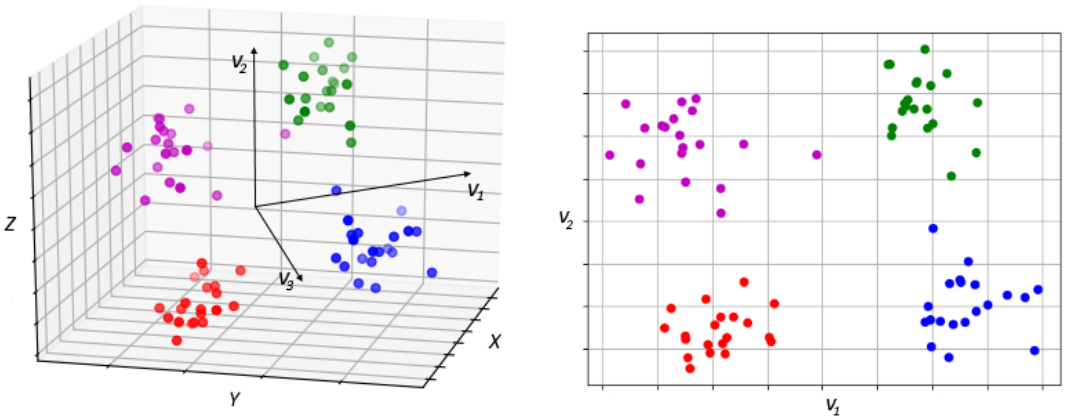
Matemaattisesti tarkasteltuna pääkomponenteiksi saadaan yhtälön (9) kovarianssimatriisin estimaatin hajontamatriisin S (engl. scatter matrix) ominaisarvoja λ vastaavat ominaisvektorit v [19]. Ominaisarvot järjestetään suuruusjärjestykseen ja suurinta ominaisarvoa vastaavasta ominaisvektorista tulee ensimmäinen pääkomponentti. Ominaisvektorien tulee toteuttaa yhtälön (10) ominaisarvo-ongelma. Laskennallisesti tehokkaampi tapa määrittää pääkomponentit on laskea normalisoidun datamatriisin singulaariarvohajotelma ja valitsemalla pääkomponenteiksi oikeanpuoleiset singulaarivektorit.

$$S = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T \tag{9}$$

$$Sv = \lambda v, \tag{10}$$

missä S on hajontamatriisi,
x_i on yksittäinen datanäyte,
$\bar{x}$ on datan otoskeskiarvo,
v on ominaisarvomatriisia vastaava ominaisvektorimatriisi ja
λ on ominaisarvot diagonaalimatriisissa.

Pääkomponenttianalyysi toimii hyvin, mikäli muuttujien välillä on vahvoja korrelaatioita ja datan riippuvuus on lineaarisesti selitettävissä. Se ei kuitenkaan onnistu muuntamaan hyvin dataa, joka sisältää paljon epälineaarisia riippuvuuksia. Kuvassa 11 (a) on eräs normaalijakautunut kolmidimensioinen data. Kolmiulotteiseen data-avaruuteen on laskettu pääkomponentit v_1, v_2 ja v_3, jonka jälkeen pääkomponenteista v_1, v_2 tulee kaksiulotteisen kuvaajan akselit kuvan 11 (b) mukaisesti ja dimensioita saadaan pudotettua yhdellä.



(a) (b)

Kuva 11. Kolmidimensioisen datan muunnos kaksidimensioiseksi pääkomponenttianalyysillä.

3.2. Monidimensioskaalaus

Monidimensioskaalaus (engl. multidimensional scaling, MDS) on joukko menetelmiä, joissa lasketaan aluksi korkeaulotteisessa avaruudessa näytteiden väliset etäisyydet, jonka jälkeen näytteet projisoidaan matalampidimensioiseen avaruuteen siten, että etäisyydet säilyvät muunnoksen jälkeen mahdollisimman ennallaan. MDS ottaa datan sisään symmetrisenä etäisyysmatriisina, jossa matriisin elementit kuvaavat näytteiden erilaisuutta/etäisyyttä toisiinsa nähden.

Jos näytteiden erilaisuuden mittana käytetään euklidista etäisyyttä, tuottavat MDS ja pääkomponenttianalyysi saman lopputuloksen. Tällöin MDS:stä käytetään nimitystä klassinen monidimensioskaalaus (engl. classical multidimensional scaling), joka on lineaarinen dimensionaalisuuden vähentämismenetelmä. Usein kuitenkin näytteiden erilaisuutta mitataan jollain muulla etäisyysfunktiolla datan epälineaaristen riippuvuussuhteiden löytämiseksi.

Täydellistä sovitusta eri dimensioiden välillä ei voida saavuttaa mutta käyttämällä yhtälön (11) minimoitavaa häviöfunktiota (engl. stress function) voidaan saada hyviä tuloksia.

$$\min \frac{1}{g} \sum_{i<j}^{N} \frac{1}{w} (\delta_{ij} - d_{ij})^2, \tag{11}$$

missä g on normalisointikerroin,
w on painokerroin,
δ_{ij} on datan näytteiden i ja j välinen etäisyys korkeassa ulottuvuudessa ja
d_{ij} on datan näytteiden i ja j välinen etäisyys matalassa ulottuvuudessa.

Normalisointikerrointa g käytetään poistamaan neliöityjen virhevektoreiden riippuvuus etäisyyden skaalasta ja painokertoimella w painotetaan näytteiden välistä erilaisuuden mittaa.

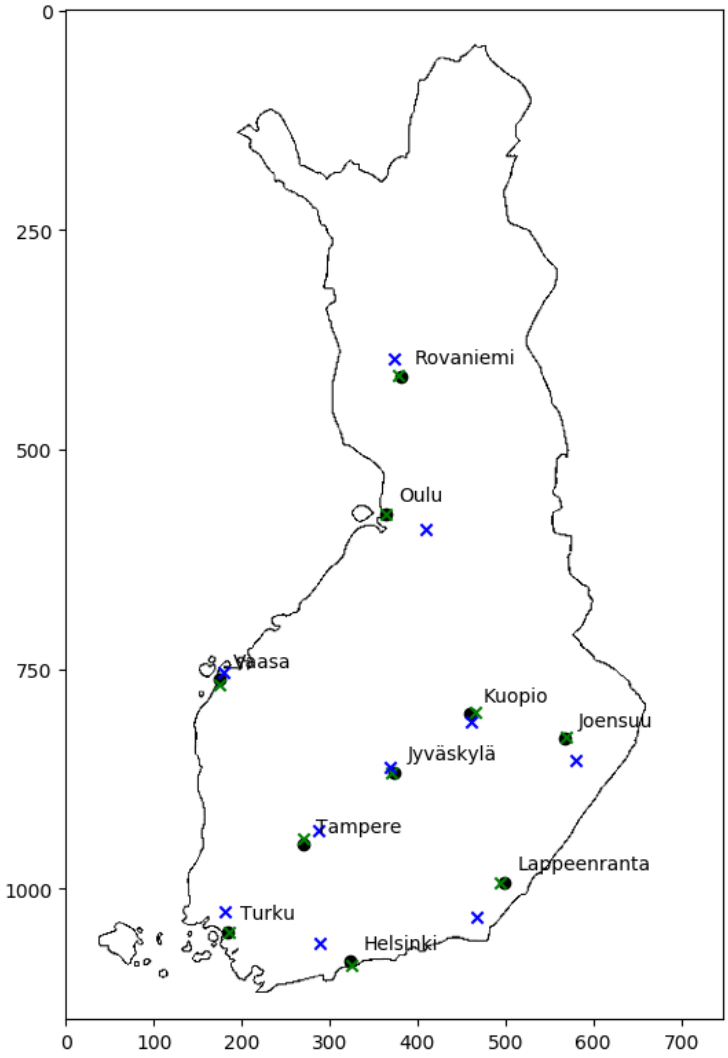
Taulukossa 3 on kolme erilaista häviöfunktiota esitettynä muuttujien g ja w avulla. Yksinkertaisin näistä on mitallinen häviöfunktio, joka minimoi dimensioiden välisten etäisyyksien neliöityjen virhevektorien summan ilman normalisointia tai painotusta [20]. Sammonin kuvaus on epälineaarinen menetelmä, jossa minimoitava häviöfunktio antaa suuremman painoarvon lyhyemmille etäisyyksille kuin pidemmille [21]. Näin saadaan säilytettyä paremmin datan paikallinen rakenne verrattuna esimerkiksi klassiseen monidimensioskaalaukseen. Epämitallinen Kruskalin häviöfunktio puolestaan keskittyy näytteiden läheisyydestä laskettaviin sijalukuihin, joita sovitetaan dimensioiden välillä [22].

Tarkastellaan esimerkkinä MDS:n toiminnasta mitallisen häviöfunktion ja epämitallisen Kruskalin häviöfunktion vaikutuksia. Datana käytetään kymmenen Suomen kaupungin maantieteellisiä etäisyyksiä toisistaan, jotka ovat taulukossa 4 symmetrisenä etäisyysmatriisina. Tämän jälkeen muunnetaan kymmendimensioinen data MDS:llä kaksidimensioiseksi. Lopuksi skaalataan muunnetut datanäytteet Suomen kartalle kuvan 12 mukaisesti. Kuvassa mustat pisteet kuvaavat kaupunkien todellisia sijainteja kartalla, vihreät rastit mitallisella häviöfunktiolla laskettuja sijainteja ja siniset rastit Kruskalin häviöfunktiolla laskettuja sijainteja.

Koordinaattiakselien mittayksikkönä käytetään kilometriä. Kuvasta havaitaan, että etäisyydet kaupunkien välillä ovat säilyneet lähes ennallaan mitallisella häviöfunktiolla mutta Kruskalin häviöfunktion antama lopputulos on paljon suurpiirteisempi.

Taulukko 3. MDS:n eri häviöfunktioiden normalisointi- ja painokertoimia

Kuormitusfunktio	g	w
Mitallinen häviöfunktio	1	1
Sammonin häviöfunktio	$\sum_{i<j}^{N} \delta_{ij}$	δ_{ij}
Kruskalin häviöfunktio	$\sum_{i<j}^{N} \delta_{ij}^2$	1



Kuva 12. MDS:n tuottamat lopputulokset kaupunkien sijainneiksi Suomen kartalla eri häviöfunktioilla.

Taulukko 4. Kymmenen Suomen kaupungin maantieteelliset etäisyydet toisistaan symmetrisessä etäisyysmatriisissa

	Helsinki	Joensuu	Jyväskylä	Kuopio	Lappeenr.	Oulu	Rovaniemi	Tampere	Turku	Vaasa
Helsinki	0	373	235	336	203	540	706	161	150	370
Joensuu	373	0	211	111	191	341	475	336	464	417
Jyväskylä	235	211	0	123	184	309	474	133	272	231
Kuopio	336	111	123	0	206	259	412	255	394	307
Lappeenr.	203	191	184	206	0	461	618	241	329	411
Oulu	540	341	309	259	461	0	166	400	533	284
Rovaniemi	706	475	474	412	618	166	0	565	695	426
Tampere	161	336	133	255	241	400	565	0	142	210
Turku	150	464	272	394	329	533	695	142	0	296
Vaasa	370	417	231	307	411	284	426	210	296	0

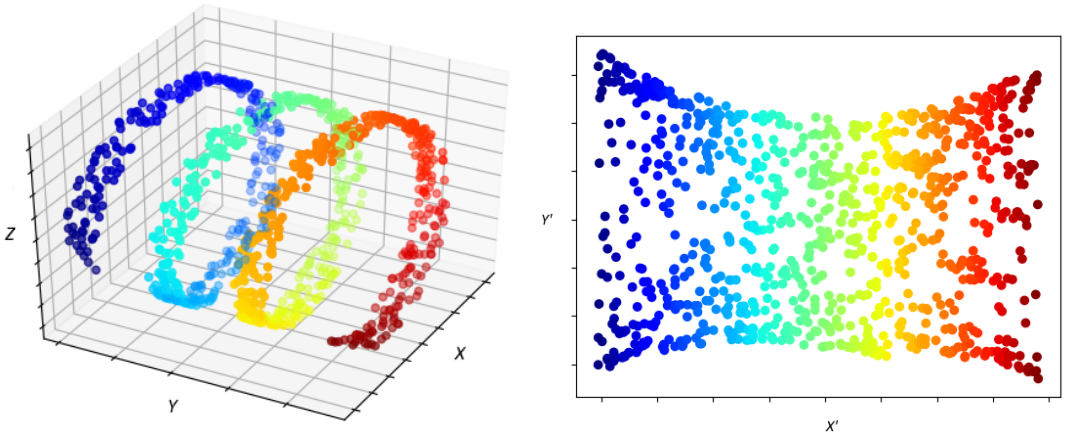
3.3. Isomap

Monidimensioskaalausmenetelmiä käyttämällä saadaan tuotettua hyödyllisiä muunnoksia monille eri datoille. Niiden pyrkiessä säilyttämään ainoastaan datanäytteiden väliset euklidiset etäisyydet, ei huomioon oteta ollenkaan datan jakaumaa tai monikertaa (engl. manifold). Esimerkiksi datanäytteet saattavat sijaita kaarevalla monikerralla kuvan 13 (a) tapaan. Tällöin MDS luulee kahta datanäytettä vierekkäisiksi näytteiksi, vaikka niiden etäisyys monikerralla olisi reilusti suurempi kuin niiden välinen euklidinen etäisyys.

Epälineaarinen dimensionaalisuuden vähentämismenetelmä Isomap säilyttää datanäytteiden väliset geodeettiset etäisyydet eli näytteiden väliset etäisyydet monikerralla [23]. Geodeettisten etäisyyksien laskemiseksi rakennetaan aluksi naapuruusgraafi liittämällä jokainen datanäyte k-lähimpään naapuriinsa. Seuraavaksi graafissa estimoidaan pisteiden välisiä lyhimpiä etäisyyksiä esimerkiksi Dijkstran algoritmilla tai Floyd–Warshall algoritmilla. Lopuksi geodeettiset etäisyydet sijoitetaan etäisyysmatriisiin, joka syötetään klassisen monidimensioskaalausalgoritmin ratkaistavaksi.

Koska lasketut geodeettiset etäisyydet ovat vain estimaatteja todellisista geodeettisista etäisyyksistä, saattaa Isomap luoda virheellisiä polkuja naapuruusgraafissa. Mikäli datanäytteet ovat jossain kohtaa monikertaa harvassa, yliarvioi Isomap geodeettisia etäisyyksiä, mikä näkyy liian venyneinä kohtina tuotetussa muunnoksessa.

Kuvassa 13 (b) on tehty kolmidimensioisen korkkiruuvinmuotoisen monikerran muunnos kaksidimensioiseksi Isomap-algoritmilla. Vierekkäisiä näytteitä monikerralla on havainnollistettu asteittain muuttuvin sateenkaaren värein. Lopputuloksen perusteella monikerran muoto on onnistuttu säilyttämään hyvin ja kuvaus on onnistunut.



(a) (b)

Kuva 13. Kolmedimensioisen korkkiruuvinmuotoisen monikerran muunnos kaksidimensioiseksi Isomap-algoritmilla.

3.4. LLE

LLE (engl. local linear embedding) algoritmi pyrkii löytämään datan epälineaariset paikalliset rakenteet säilyttäen naapurinäytteiden väliset etäisyydet monikerralla muunnetussa dimensiossa [24]. Perinteinen LLE koostuu kolmesta vaiheesta. Aluksi jokaiselle datanäytteelle etsitään k-lähimmät naapurit Isomap-algoritmin tapaan geodeettisten etäisyyksien laskemiseksi. Toisessa vaiheessa lasketaan painokertoimet jokaiselle k-lähimmälle naapurinäytteelle minimoiden yhtälön (12) häviöfunktio. Optimoituja painokertoimia käyttämällä muodostetaan naapurinäytteistä lineaarikombinaatio, joka kuvaa parhaiten tutkittavaa näytettä. Painokertoimia rajoittavat kaksi ehtoa: 1) Mikäli näyte x_j ei kuulu näytteen x_i k-lähimpään naapuriin, tulee painokertoimen w_{ij} olla 0. 2) Näytteen x_i naapurinäytteiden x_j painokertoimien summan $\sum_{j=1}^{M} w_{ij}$ tulee olla 1.

$$\min \sum_{i=1}^{N} |x_i - \sum_{j=1}^{M} w_{ij} x_j|^2, \tag{12}$$

missä x_i on tutkittava datanäyte korkeassa ulottuvuudessa,
x_j kuuluu datanäytteen x_i k-lähimpään naapuriin korkeassa ulottuvuudessa ja
w_{ij} on painokerroin.

Viimeisessä vaiheessa jokaiselle korkeadimensioiselle näytteelle x_i haetaan mataladimensioinen esitystapa y_i. Tämä tapahtuu minimoimalla yhtälön (13) häviöfunktio. Nyt painokertoimet w_{ij} tiedetään yhtälössä (12) tehdyn optimoinnin perusteella ja voidaan laskea paras rekonstruktio näytteille y_i matalassa ulottuvuudessa.

$$\min \sum_{i=1}^{N} |y_i - \sum_{j=1}^{M} w_{ij} y_j|^2, \tag{13}$$

missä y_i on muunnettu datanäyte matalassa ulottuvuudessa ja
y_j kuuluu datanäytteen y_i k-lähimpään naapuriin matalassa ulottuvuudessa.

3.5. Itseorganisoiva kartta

Itseorganisoiva kartta (engl. self-organizing map, SOM) on Teuvo Kohosen kehittämä ohjaamattoman oppimisen neuroverkkomenetelmä, jota käytetään monidimensioisen datan visualisoimiseen [25]. Itseorganisoiva kartta sisältää N kappaletta kaksiulotteiseen tasoon sijoitettua neuronia, jossa jokaisella on oma painokertoimensa. Painokertoimet muodostuvat painovektorin m_i, jossa painokertoimien lukumäärä vastaa korkeadimensioiden datan muuttujien lukumäärää. Seuraavaksi iteratiivisessa opetusvaiheessa valitaan jokaiselle näytteelle x yksi kerrallaan parhaiten sopiva neuroni. Parhaiten sopivaksi neuroniksi valitaan yhtälön (14) mukaisesti pienimmän painovektorin m_i ja datanäytteen x välisen etäisyyden saanut neuroni.

$$m_c = \arg\min_i \ |x - m_i|, \tag{14}$$

missä m_c on parhaiten sopiva painovektori,
x on korkeadimensioinen datanäyte ja
m_i on painokertoimista koostuva painovektori.

Tämän jälkeen painovektoreita päivitetään yhtälön (15) päivityssäännöllä vastaamaan paremmin syötettävää dataa.

$$m_i(t+1) = m_i(t) + \alpha(t) h_{ci}(t)[x(t) - m_i(t)], \tag{15}$$

missä t on aikayksikkö,
α on oppimisnopeus ja
h_{ci} on naapuruusfunktio.

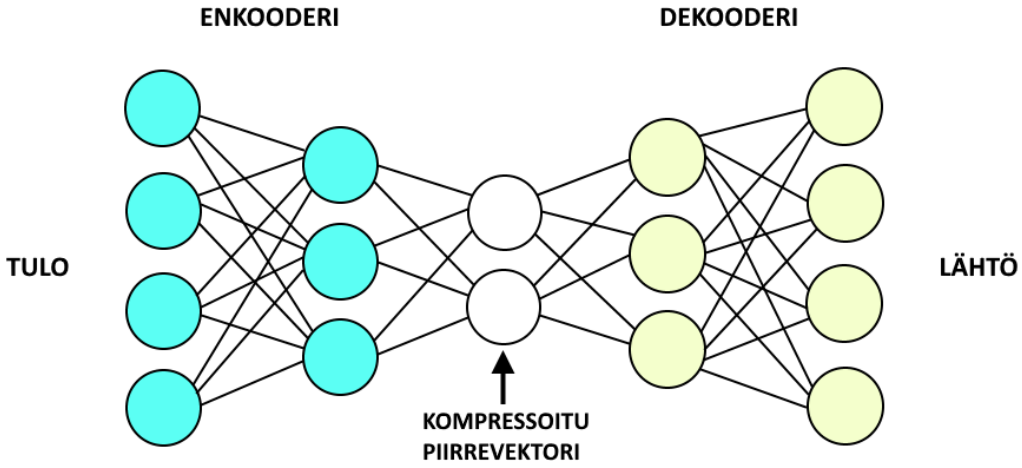
Oppimisnopeus t kertoo missä määrin uusi tieto korvaa vanhan tiedon. Naapuruusfunktio h_{ci} arvottaa sovituksen voimakkuutta ja siihen vaikuttavien neuronien lukumäärää. Oppimisnopeutta ja naapuruusfunktiota on tapana pienentää opetuksen edetessä.

Itseorganisoiva kartta onnistuu löytämään datan epälineaarisia riippuvuussuhteita. Opetettu itseorganisoiva kartta oppii opetusdatan todennäköisyystiheysjakauman ja approksimaation, mihin kohtaa karttaa eri opetusdatan ominaisuudet heijastuvat. Uusi testinäyte kuvautuu siis siihen neuroniin kartassa, joka on sitä lähimpänä opitussa piirreavaruudessa.

3.6. Autoenkooderi

Autoenkooderi on neuroverkkomalli, joka oppii datan rakenteisuuden ja esitystavan ohjaamattomasti [26]. Sitä käytetään yleisimmin dimensionaalisuuden vähentämisessä tehokkaan piirteiden esitystavan oppimiseksi, mutta myös datan visualisoinneissa. Neuroneista koostuva autoenkooderi käsittelee siihen tulevaa dataa enkooderissa kompressoiden datanäytteen piirrevektoriksi kuvan 14 tapaan. Enkooderin lähtö syötetään dekooderille, joka rekonstruoi kompressoiduista piirteistä alkuperäisiä datan dimensioita vastaavan lähdön. Neuroverkon opettamisen jälkeen dekoodausosa jätetään pois ja pystytään ennustamaan uusille tuntemattomille datanäytteille kompressoidut piirteet.

Autoenkooderit voidaan toteuttaa monikerros-perceptroni-rakenteina tai konvoluutioneuroverkkoina. Nykyään autoenkoodereita käytetään paljon generatiivisissa malleissa, joissa opitusta piirreavaruudesta generoidaan uusia näytteitä. Tästä hyvä esimerkki on muuntuva autoenkooderi, joka oppii datan todennäköisyystiheysjakauman estimaatin uusien näytteiden generoimiseksi.



Kuva 14. Neuroneista koostuva autoenkooderi.

3.7. t-SNE

t-SNE (engl. t-distributed stochastic neighbor embedding) on vuonna 2008 julkaistu epälineaarinen datan visualisointi ja dimensionaalisuuden vähentämismenetelmä [27]. Se pyrkii säilyttämään datan lokaalin rakenteen eli samanlaiset näytteet alkuperäisessä data-avaruudessa ovat muunnoksen jälkeen mahdollisimman lähellä toisiaan. Toisaalta t-SNE pyrkii säilyttämään myös datan globaalin rakenteen eli erilaiset klusterit alkuperäisessä data-avaruudessa ovat muunnoksen jälkeen mahdollisimman erillään toisistaan.

t-SNE-algoritmi muodostaa aluksi todennäköisyystiheysjakauman jokaiselle näytteelle erikseen korkeassa ulottuvuudessa siten, että samanlaiset naapurit saavat mahdollisimman suuren arvon ja erilaiset naapurit mahdollisimman pienen. Seuraavaksi t-SNE-algoritmi määrittää korkean ulottuvuuden kanssa samankaltaiset

todennäköisyystiheysjakaumat matalassa ulottuvuudessa sattumanvaraisesti sijoitetuille näytteille. Lopuksi algoritmi etsii näytteille optimaaliset paikat matalassa ulottuvuudessa minimoiden iteratiivisesti Kullback-Leibler divergenssin. Päämääränä on, että lopputuloksena näytteiden todennäköisyystiheysjakaumat vastaavat mahdollisimman hyvin toisiaan ulottuvuuksien välillä. Yleisimmin algoritmissa käytetään samankaltaisuuden mittana euklidista etäisyyttä, mutta sen sijasta voidaan käyttää myös muita painotettuja etäisyysmittoja.

t-SNE-algoritmilla pystytään analysoimaan ja visualisoimaan mitä tahansa korkeadimensioista dataa, mutta erityisesti sitä on käytetty mm. lääketieteellisessä kuvantamisessa [28], kasvonpiirteiden tunnistuksessa [29] sekä puheen tunnistuksessa [30].

3.8. UMAP

UMAP (engl. uniform manifold approximation and projection) on vuonna 2018 julkaistu epälineaarinen dimensionaalisuuden vähentämismenetelmä [31]. Se pyrkii oppimaan datan monikerran rakenteen ja hakee sille mataladimensioisen esitystavan säilyttäen monikerran keskeisimmät topologiset ominaisuudet.

Aluksi UMAP-algoritmi muodostaa korkeassa dimensiossa datan näytteille sumean topologisen esitystavan. Algoritmi olettaa, että data on jakautunut tasaisesti Riemannin monikerran päälle sekä Riemannin metriikka on lokaalisti vakioarvoinen. Tämä toteutuu määrittämällä jokaisen datanäytteen ympärille k-lähimpään naapurin ulottuva sumean logiikan luottamusvälinen hyperpallo. Lisäksi monikerran tulee koostua lokaaleista konnektioista. Tämä ehto täyttyy, kun lähimmät naapurit korkeassa dimensiossa yhdistetään sumeiksi simplekseiksi, joista muodostuu painotettu graafi. Lopuksi etsitään vastaavanlainen sumea topologinen esitystapa matalassa dimensiossa ja minimoidaan yhtälön (16) esitystapojen välinen ristientropia. Minimoitavan summan ensimmäinen termi muodostaa puoleensa vetävän voiman ja toinen termi hylkivän voiman. Tasapainottelemalla näiden vetävien ja hylkivien voimien välillä löydetään esitystapa, joka kuvaa hyvin datan topologiaa.

$$\min \sum_{i=1}^{N} w_{Hi} log(\frac{w_{Hi}}{w_{Li}}) + (1 - w_{Hi}) log(\frac{1 - w_{Hi}}{1 - w_{Li}}), \tag{16}$$

missä w_{Hi} on 1-simpleksin painokerroin korkeassa ulottuvuudessa ja
w_{Li} on 1-simpleksin painokerroin matalassa ulottuvuudessa.

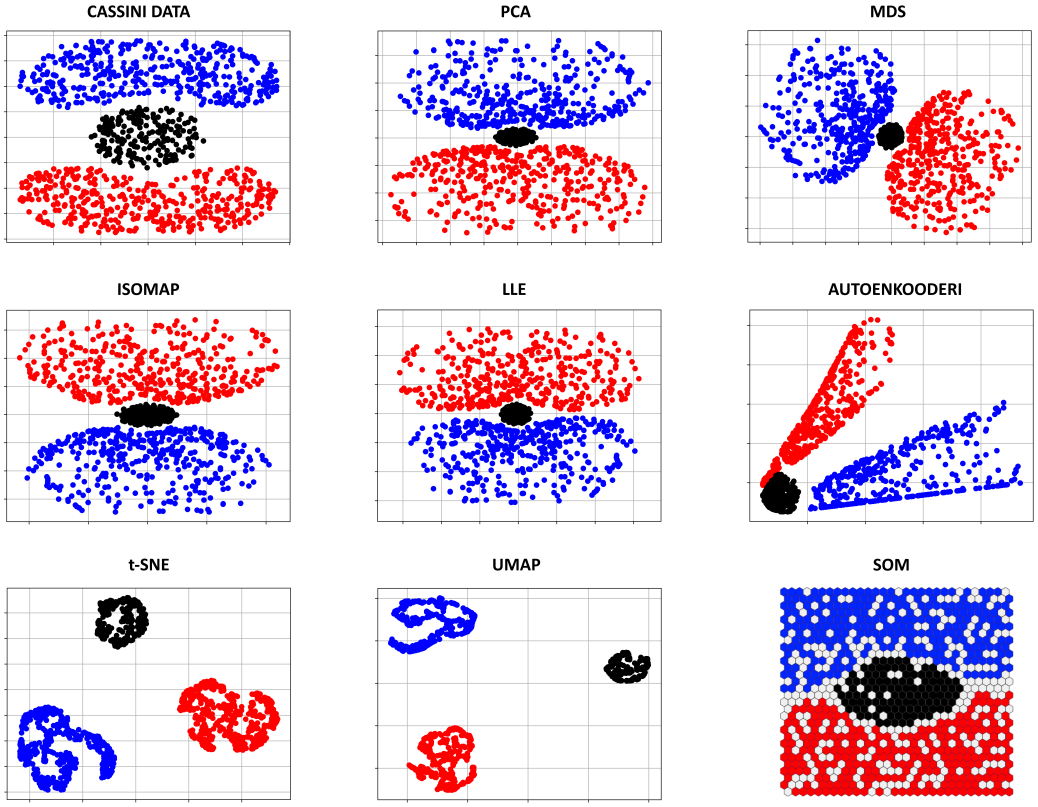
3.9. Dimensionaalisuuden vähentäminen datan visualisoinneissa

Datan visualisoimista käytetään selittävän data-analyysin keinona, jossa pyritään näkemään datan sisälle ja selvittämään datan ominaisuuksia sen rakenteen perusteella. Datan visualisointien tavoitteena on säilyttää datan sisältämä informaatio dimensionaalisuuden vähentämisen jälkeisessä, ihmiselle helposti

ymmärrettävässä kaksi- tai kolmiulotteisessa avaruudessa. Eri dimensionaalisuuden vähentämismenetelmät lähestyvät ongelmaa eri näkökulmista ja pyrkivät säilyttämään datan eri ominaisuuksia. Datan visualisointien lopputulosten onnistumista on haastavaa mitata, mutta niitä voidaan arvioida empiiristen havaintojen perusteella. Toinen vaihtoehto onnistumisen mittaamiseksi on opettaa yksinkertainen luokittelija alkuperäisellä ja muunnetulla datalla ja vertailla, esimerkiksi kuinka paljon luokittelutarkkuus muuttuu dimensioiden välillä.

Monia dimensionaalisuuden vähentämismenetelmiä ei ole alun perin suunniteltu datan visualisoimista varten, vaan tiivistämään dataa päästen eroon tarpeettomista piirteistä. Esimerkiksi pääkomponenttianalyysia käytetään yleisimmin datan esikäsittelytekniikkana datan visualisoimisen sijaan.

Eri dimensionaalisuuden vähentämismenetelmillä tuotetut visualisoinnit vaihtelevat suuresti. Tunnetussa Cassini-data esimerkissä keinotekoinen kaksidimensioinen tuhannen näytteen data muunnetaan yhdeksän dimensioiseksi muunnoksella $(x + y, x - y, xy, x^2, y^2, x^2y, xy^2, x^3, y^3)$ [32]. Cassini-datassa on kolme tasajakautunutta klusteria, joista kaksi isompaa 400 näytteen klusteria ympäröi 200 näytteen keskimmäistä klusteria. Kuvassa 15 muunnoksen jälkeinen yhdeksän dimensioinen data on muunnettu takaisin kaksidimensioisiksi visualisoinneiksi aiemmin esitetyillä dimensionaalisuuden vähentämismenetelmillä.



Kuva 15. Cassini-datan muunnoksen visualisointeja eri dimensionaalisuuden vähentämismenetelmillä.

Visualisoinneista nähdään, kuinka algoritmit pääkomponenttianalyysi, monidimensioskaalaus, Isomap, LLE ja itseorganisoiva kartta säilyttävät klustereiden rakenteet, mutta klustereiden välinen luokkaraja on lähes huomaamaton.

Itseorganisoiva kartta värjää kuusikulmioista koostuvan data-avaruuden sen mukaan, minkä luokan näytteitä kuusikulmioiden kohdalla esiintyy eniten. Algoritmit t-SNE ja UMAP ovat erottaneet klusterit selvästi erilleen toisistaan, mutta eivät ole onnistuneet täysin säilyttämään klustereiden näytteiden välisiä sisäisiä riippuvuuksia. Selvästi huonoiten visualisoimisessa onnistui autoenkooderi, joka projisoi punaisen ja mustan klusterin näytteitä päällekkäin. Huomioitavaa on, että jos data olisi reilummin epäbalansoitu, vaikuttaisi se mm. itseorganisoivan kartan tuottamaan lopputulokseen negatiivisesti.

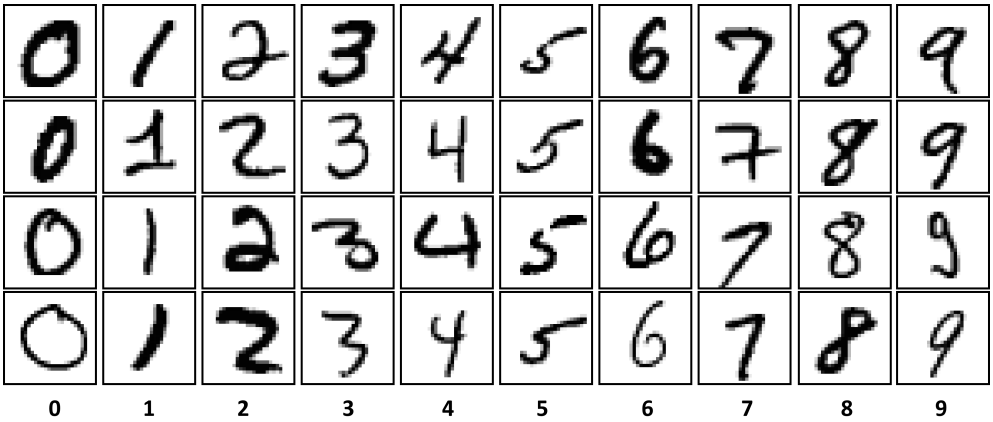
4. DATAN VISUALISOIMINEN

Tässä luvussa esitellään uudenlainen aktiivisen oppimisen lähestymistapa luokkatiedottomien näytteiden kategorisoimiseksi. Visualisoimalla korkeadimensioinen data kaksiulotteisina esityksinä, pystyy oppimisprosessiin osallistuva ihminen muokkaamaan helpommin luokkarajoja. Tämä tapahtuu manuaalisesti valitsemalla kategorisoitavaksi luokkarajojen lähellä olevia näytteitä ja pudottamalla opetusdatasta kokonaan pois harhaanjohtavia näytteitä. Datan visualisointityökalu tarjoaa helppokäyttöisen käyttöliittymän aktiiviseen oppimisprosessiin osallistuvan ihmisen kategorisoimistyön helpottamiseksi.

4.1. Tutkimuksessa käytetty data

Datan visualisoimistyökalun suorituskyvyn testaamisessa käytettiin kolmea erilaista dataa: MNIST-dataa, sahatavaran vioista koostuvaa oksadataa [33] sekä loiseläinten munista ja alkueläinten kystista kerättyä dataa[1]. Vaikka MNIST-datalla on helppo demonstroida kategoriapäätösten vaikutusta opetettavaan malliin, on se luokitteluongelmana yksinkertainen. Tämän takia tutkimukseen otettiin mukaan kaksi monimutkaisempaa, sovelluksille tyypillistä dataa aktiivisen oppimisen mallien oppimisprosessien arvioimisessa.

MNIST-data sisältää käsinkirjoitetut numerot 0:sta 9:sään. Yhteensä balansoidussa MNIST-datassa on 70 000 näytettä, jotka on kerätty Yhdysvaltojen väestölaskentaviraston työntekijöiltä ja toisen asteen opiskelijoilta. MNIST-datanäytteet on skaalattu 20x20 pikselin mustavalkokuviksi ja lopuksi keskitetty 28x28 pikselin ruudukolle. Tällä datalla on onnistuttu opettamaan syväoppimisen menetelmillä luokittelija, jonka luokittelutarkkuus on 99,79 % [34]. Kuvassa 16 on esimerkkinäytteitä MNIST-datan kymmenestä luokasta.



Kuva 16. Esimerkkinäytteitä MNIST-datan kymmenestä luokasta.

Toisena datana tutkimuksessa käytettiin sahatavaran vikojen yleisimpiä oksatyyppejä. Oksa on kovaa puuainesta, jonka syysuunta on kohtisuorassa puun rungon syysuuntaan. Oksadatan RGB-kuvat on skaalattu 100x100 pikselin kokoon.

[1] https://github.com/tholmb/worm-egg-data

Oksien tyypit voidaan jakaa luokkiin oksan laadun ja sahaussuunnan perusteella. Taulukossa 5 on alkuperäisen oksadatan luokat ja näytteiden lukumäärät luokkaa kohden. Peilaamalla näytteitä x-akselin ja y-akselin suuntaisesti saadaan niiden lukumäärää kasvatettua neljä kertaa suuremmaksi kokeiluja varten. Kuvassa 17 on esimerkkikuvia oksadatan luokista.

Taulukko 5. Alkuperäisen oksadatan luokat ja näytteiden lukumäärät luokkaa kohden

Luokka	Näytteiden lukumäärä
Tuoreoksa	179
Kuiva oksa	69
Särmäoksa	65
Lehtioksa	47
Sarvioksa	35
Kuoren ympäröimä oksa	29
Laho oksa	14

Kuva 17. Esimerkkinäytteitä oksadatan luokista.

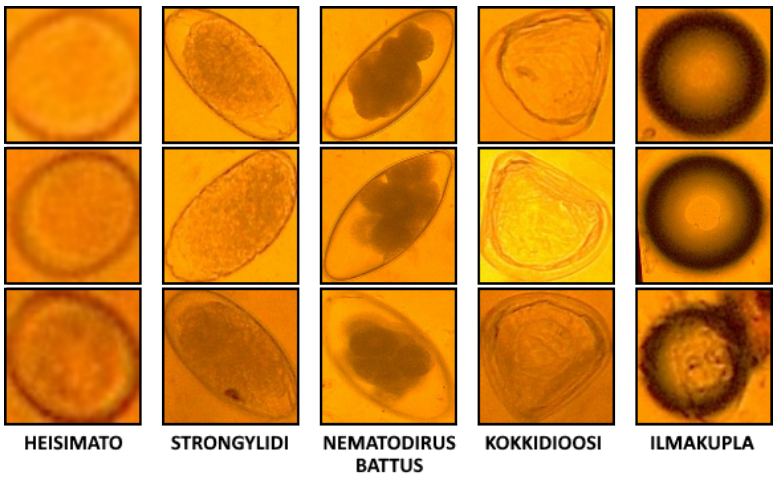
Tutkimuksen yhteydessä kerättiin loiseläinten madonmunista ja alkueläinten kystista koostuva data. Eläimen parasiittien munien tyypistä ja lukumäärästä ulosteessa voidaan päätellä tarvittava lääkeaine ja sen määrä loisinfektioiden hoitamiseksi. Data on kerätty kuvaamalla madonmunien tutkimiseen ja lukumäärän laskemiseen käytetyn McMaster-laskentakammion pintaa mikroskooppikameralla 10 kertaisesti suurentavalla objektiivilla.

McMaster-laskentakammio täytettiin lampaan ulosteflotaatiosuspensiolla Suomen Ruokaviraston standardien mukaisesti. Kuvien ottamisen jälkeen niistä rajattiin ja kategorisoitiin munat eri loiseläintyypin mukaan. Madonmunien tunnistuksen kannalta on tärkeää erottaa suspensionäytteissä esiintyvät ilmakuplat omaksi luokakseen, ettei niitä sekoiteta vahingossa madonmuniksi. Lopuksi RGB-kuvat skaalattiin 100x100 pikselin kokoon. Taulukossa 6 on alkuperäisen madonmunadatan luokat ja näytteiden lukumäärät luokkaa kohden. Peilaamalla myös madonmunanäytteitä x-akselin ja y-

akselin suuntaisesti saadaan niiden lukumäärää kasvatettua neljä kertaa suuremmaksi kokeiluja varten. Kuvassa 18 on esimerkkikuvia datan luokista.

Taulukko 6. Alkuperäisen madonmunadatan luokat ja näytteiden lukumäärät luokkaa kohden

Luokka	Näytteiden lukumäärä
Heisimato	160
Strongylidi	30
Nematodirus battus	4
Kokkidioosi	29
Ilmakupla	10



Kuva 18. Esimerkkinäytteitä madonmunadatan luokista.

4.2. Datan visualisoimistyökalu

Datan visualisoimista varten kehitettiin Python-ohjelmistotyökalu, jossa käsitellään yhtä kerrallaan kolmesta tutkimukseen käytetyistä datoista. Aluksi data jaetaan opetusdataan ja testidataan käyttäjän määrittelemällä jakosuhteella. Aktiivisen oppimisen prosessin alussa on valittava, montako näytettä on alustettuna kategorisoitujen näytteiden X^L dataan. Alustettujen näytteiden lukumäärään tulee olla vähintään yhtä suuri kuin datan luokkien lukumäärä, jotta jokaisesta luokasta saadaan mallille ainakin yksi opetusnäyte. Toisaalta alustettujen näytteiden lukumäärä ei saa olla liian iso, jotta aktiivisesta oppimisesta saadaan mahdollisimman suuri hyöty.

Datan visualisoimistyökalu käyttää näytejoukkoihin perustuvaa näytteistystä, jossa kategorisoimattomien näytteiden osajoukosta osoitetaan informatiivisimmat näytteet kaksiulotteisessa kuvauksessa. Ennen opetuksen alkamista käyttäjä valitsee näytteistysstrategian osajoukon koon ja kuinka monta ao. strategian mukaisesti paremmuusjärjestykseen asetettua näytettä visualisoidaan.

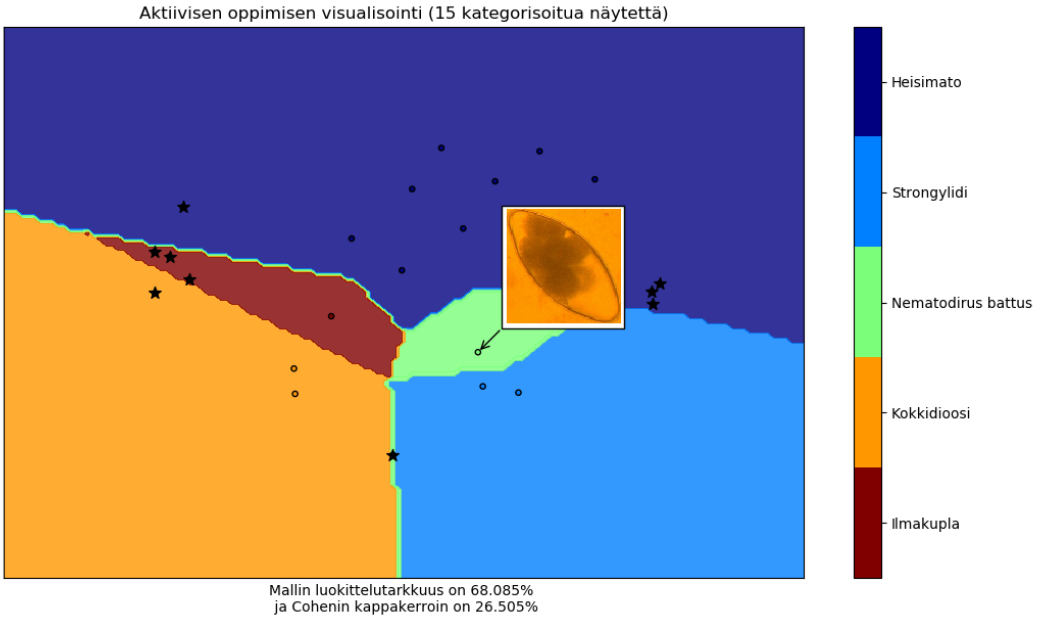
Epävarmuuteen perustuvista näytteistysstrategioista on valittavissa satunnainen näytteistys, epäluotettavuuteen perustuva näytteistys, pienimmän marginaalin

näytteistys ja maksimaalisen entropian näytteistys. Komiteakyselyssä näytteistysstrategioista voidaan valita pehmeän äänestyksen entropian, kovan äänestyksen entropian tai keskiarvoistetun Kullback-Leibler divergenssin.

Komitean mallien lukumäärä on myös tärkeä parametri, joka tulee valita huolella. Oletettavasti mitä enemmän malleja komitea sisältää, sitä varmempi se tulee olemaan luokkatiedottoman näytteen luokkien posterioritodennäköisyyksistä. Monia jäseniä sisältävien komiteoiden opettamiseen kuluu kuitenkin paljon aikaa, mikä asettaa komitean koolle ylärajan.

Käytettävät mallit tai malleista koostuvat komiteat voidaan opettaa yhdellä seuraavista scikit-learn Python-kirjaston luokittelualgoritmeista: tukivektorikone, päätöspuu, satunnaismetsä, adaptiivinen boostaus, naive bayes, lineaarinen diskriminanttianalyysi, logistinen regressio ja monikerros-perceptroni. Datan kaksiulotteisen esityksen tuottava dimensionaalisuuden vähentämismenetelmä on valittavissa seuraavista vaihtoehdoista: pääkomponenttianalyysi, monidimensioskaalaus, LLE, Isomap, t-SNE ja UMAP. Koska scikit-learn Python-kirjasto ei tarjoa monidimensioskaalauksen ja t-SNE-algoritmin muunnokselle erillistä muunnosfunktiota, opetetaan molempien menetelmien muunnoksen arvioimiseksi monen selittävän tekijän regressiomalli.

Lopuksi aktiivisen oppimisen tueksi data visualisoidaan kaksiulotteiseksi kuvan 19 tapaan, jossa kategorisoitavana on ollut madonmunadata. Visualisointityökalussa pisteillä merkityt kohdat ovat kategorisoituja näytteitä ja tähdellä merkityt kategorisoimattomia. Kuvaajan alalaitaan on merkitty mallin senhetkinen luokittelutarkkuus satunnaismetsäluokittelijalla.



Kuva 19. Madonmunadatan kategorisoimista visualisointityökalulla.

Asiantuntijan kannalta on mielenkiintoista nähdä, millaiset näytteet sijoittuvat kaksiulotteisen data-avaruuden luokkarajojen läheisyyteen verrattuna naapurinäytteisiin. Asiantuntijalle on siksi luotu mahdollisuus muokata luokkarajoja valitsemalla kategorisoitavaksi luokkarajojen lähellä olevia näytteitä. Lisäksi asiantuntija voi poistaa aiemmin kategorisoituja epäselviä näytteitä, vaihtaa väärin kategorisoitujen näytteiden luokkaa tai olla kategorisoimatta mitään tarjotuista näytteistä pyytämällä uuden osajoukon kategorisoitavaksi.

5. TULOKSET JA NIIDEN ARVIOINTI

Tässä luvussa tarkastellaan perinteisellä aktiivisella oppimisella saatuja tuloksia ja vertaillaan niitä tilanteeseen, jossa datan visualisoiminen on otettu aktiivisen oppimisen tueksi. Lisäksi arvioidaan visualisoinneista, kuinka lähelle luokkarajaa vaikeimmin tunnistettavat näytteet sijoittuvat eri aktiivisen oppimisen vaiheissa.

5.1. Tulosten arvioimiseen käytetyt mittasuureet

Yleisin koneoppimismallin suorituskyvyn arvioimiseen käytetty mitta on luokittelutarkkuus. Luokittelutarkkuus kertoo yksinkertaisesti, kuinka monta prosenttia testidatan näytteistä luokiteltiin oikeisiin luokkiin. Vaikka luokittelutarkkuuden arvoja on helppo tulkita, se ei kuitenkaan anna totuudenmukaista kuvaa luokittelijan suorituskyvystä, kun data on vahvasti epäbalanssissa. Jos esimerkiksi binäärisessä luokitteluongelmassa testidatan enemmistöluokka sisältää 90 % näytteistä ja luokittelija luokittelee kaikki testidatan näytteet enemmistöluokkaan, saavutetaan lähtötilanteessa 90 % luokittelutarkkuus, vaikkei mitään luokittelua ole vielä tapahtunut. Tätä ilmiötä kutsutaan luokittelutarkkuuden paradoksiksi.

Datan epäbalanssin huomioivia mittasuureita on monia, jotka ottavat huomioon luokittelutuloksen arvioimisessa eri seikkoja. Sensitiivisyys (engl. recall) laskee oikeiden hälytysten (engl. true positives) määrän suhteessa oikeisiin positiivisiin $\frac{TP}{TP+FN}$. Positiivinen ennustearvo (engl. precision) arvioi oikeiden hälytysten määrää kaikista positiivisiksi tunnistetuista ennusteista $\frac{TP}{TP+FP}$. F1-arvo on sensitiivisyyden ja positiivisen ennustearvon harmoninen keskiarvo. Cohenin kappakerroin suhteuttaa havaitun luokittelutarkkuuden luokittelijan oletettuun luokittelutarkkuuteen, joka saavutetaan sattumanvaraisella luokittelulla yhtälön (17) mukaisesti.

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{17}$$

missä κ on Cohenin kappakerroin,
p_0 on havaittu luokittelutarkkuus ja
p_e on oletettu lähtötaso luokittelutarkkuudelle sattumanvaraisella luokittelulla.

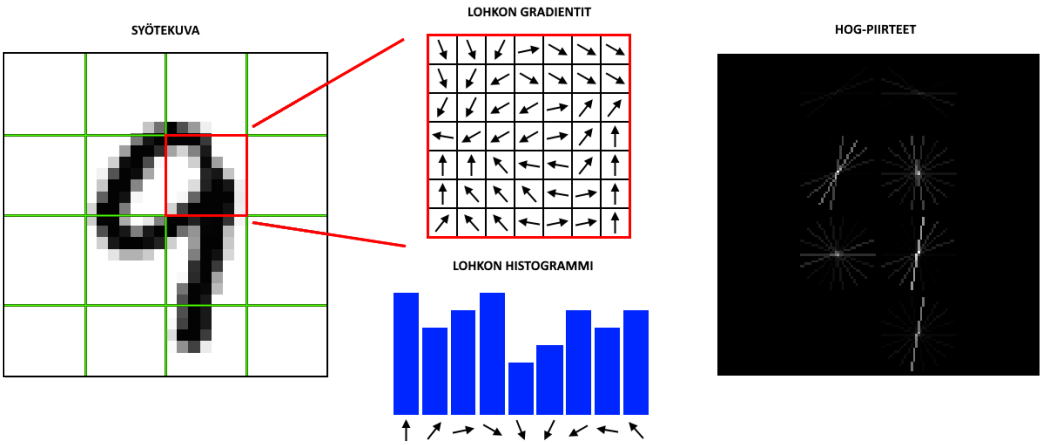
Tässä tutkimuksessa käytetään tulosten arvioimisessa balansoidulle MNIST-datalle mittasuureena luokittelutarkkuutta ja epäbalansoiduille oksadatalle ja madonmunadatalle Cohenin kappakerrointa.

5.2. Datan esikäsittely

Datan esikäsittelyssä kuvanäytteille lasketaan piirteet ennen luokittelualgoritmille syöttämistä. Piirteet kuvaavat tehokkaasti datan ominaisuuksia samalla nopeuttaen laskentaa. Yleisesti käytettyjä perinteisiä menetelmiä piirteiden irrottamiseen

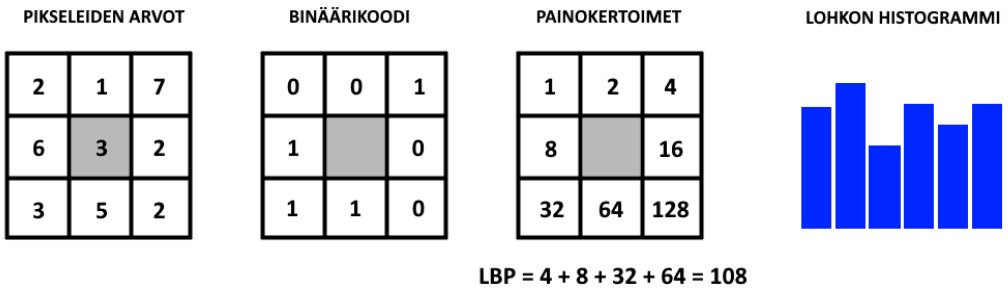
tekstuureista ovat HOG (engl. histogram of oriented gradients) [35] ja LBP (engl. local binary pattern) [36].

HOG-menetelmä laskee kuvalle piirrevektorin, joka on lähes identtinen samalle kuvalle eri kontrastin arvoilla. Aluksi menetelmässä syötekuva jaetaan lohkoihin. Jokaiselle lohkon pikselille lasketaan gradientti, joka kertoo kuvan tummenemisen suunnan ja suuruuden. Sijoittamalla lohkon gradientit histogrammiin, saadaan lohkon HOG-piirteet. Kuvassa 20 on laskettu MNIST-datanäytteelle HOG-piirteet ja lopuksi visualisoitu ne lähtökuvassa. Syötekuva on jaettu 16 lohkoon ja gradientit on kynnystetty 40 asteen välein.



Kuva 20. HOG-piirteiden laskenta MNIST-datanäytteelle.

LBP on tehokas menetelmä tekstuurien piirteiden irrottamiseen syötekuvasta. Aluksi syötekuva jaetaan lohkoihin ja ne käydään läpi konvoluutiotyyppisesti 3x3 kokoisina liukuvina maskeina. Maskin keskimmäisen pikselin perusteella lasketaan binäärikoodi, jossa keskimmäistä pikseliä pienemmät arvot saavat arvon 0 ja loput arvon 1. Kertomalla binäärikoodi painokertoimilla saadaan maskin keskimmäiselle pikselille LBP-arvo. Lopuksi lohkon LBP-arvoista muodostetaan histogrammi kuvan 21 mukaisesti.



Kuva 21. LBP-piirteiden laskenta 3x3 kokoiselle kuvalle.

Aluksi selvitettiin, mitkä piirteet toimivat parhaiten kullekin datalle. Tätä varten määritettiin luokittelutarkkuudet k-lähimmän naapurin luokittelijalla, satunnaismetsällä ja logistisella regressiolla MNIST-datalle ja Cohenin kappakertoimet oksa- ja madonmunadatalle HOG-piirteistä, LBP-piirteistä ja pikseliarvoista. Tulokset on kerätty taulukkoon 7.

Satunnaismetsälle alkuperäinen kuva tuotti kaikille datoille HOG-piirteitä ja LBP-piirteitä paremmat tulokset. Ainoastaan madonmunadata luokiteltiin k-lähimmän naapurin luokittelijalla paremmin HOG- ja LBP-piirteillä. Logistiselle regressiolle HOG-piirteillä saatiin parempi luokittelutarkkuus kuin opetettaessa MNIST-datan harmaasävykuvilla. Pääpiirteittäin tutkimuksessa käytetyille datoille parhaat luokittelutulokset saavutettiin alkuperäisillä kuvilla. Piirteiden muunnoksen yhteydessä menee selvästi tärkeää informaatiota hukkaan, minkä takia alkuperäiset kuvat syötetään suoraan luokittelijoille ilman datan esikäsittelyä.

Taulukko 7. Datan esikäsittelytekniikoiden vertailu eri datoille.

	MNIST (Luokittelutarkkuus)			Oksadata (Cohenin kappakerroin)			Madonmunadata (Cohenin kappakerroin)		
	GRAY	HOG	LBP	RGB	HOG	LBP	RGB	HOG	LBP
RF	0,9678	0,9513	0,8863	0,7581	0,6435	0,5482	0,8972	0,8523	0,7515
k-NN	0.9705	0.9220	0.8886	0.7506	0.6077	0.5444	0.7050	0.7208	0,7799
LR	0,9188	0.9430	0.8205	0.6187	0.6160	0.5993	0.9410	0.9397	0,8569

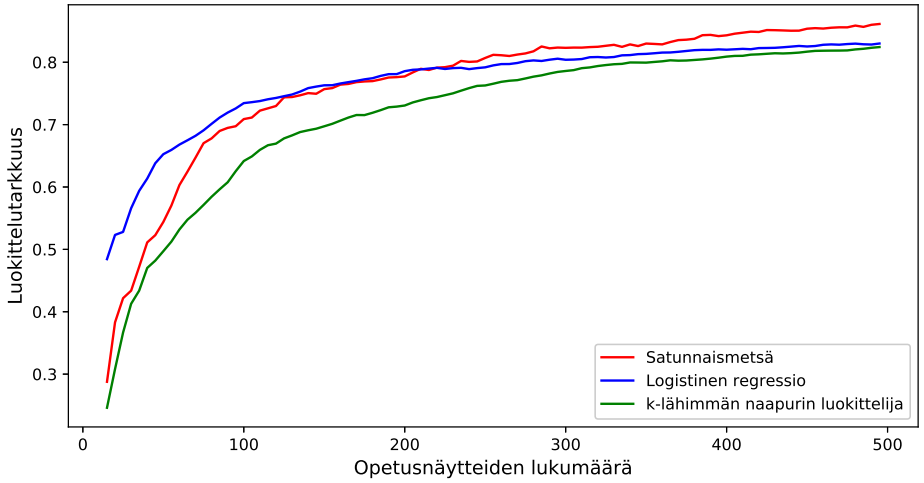
On syytä huomata, että tarkastelussa ei ole mukana piirteiden irrotukseen käytettyjä neuroverkkomenetelmiä, sillä ne ovat herkkiä hyperparametreille. Lisäksi neuroverkkojen opettaminen vie paljon aikaa, mikä on ristiriidassa aktiivisen oppimisen perusajatuksen kanssa. Toisaalta tämän tutkimuksen päätarkoitus ei ole löytää optimaalisia piirteitä vaan kiinnostavampaa on selvittää eri aktiivisen oppimisen näytteistystapojen oppimisnopeuserot.

5.3. Perinteinen aktiivinen oppiminen MNIST-datalle

Seuraavaksi vertaillaan MNIST-datalle eri luokittelijoiden luokittelutarkkuuksia satunnaisella näytteistyksellä. Koska aktiivisessa oppimisessa malli opetetaan uudestaan uusia näytteitä lisättäessä opetusdataan, tulee koneoppimisen algoritmien olla laskennallisesti kevyitä. Pienentääkseen sattuman vaikutusta, mallit opetetaan viisi kertaa ja otetaan oppimiskäyristä keskiarvot. Kuvassa 22 on vertailtu luokittelijoiden oppimista satunnaisella näytteistyksellä MNIST-datalle. Oppimisprosessin alkutilanteessa logistisen regression luokittelijalla saavutettiin paras luokittelutarkkuus. Lisättäessä opetusdataan kategorisoituja näytteitä 500 opetusnäytteen kohdalla oli satunnaismetsä paras luokittelija. Tähän tulokseen perustuen MNIST-datalle käytetään jatkossa luokittelijana satunnaismetsää.

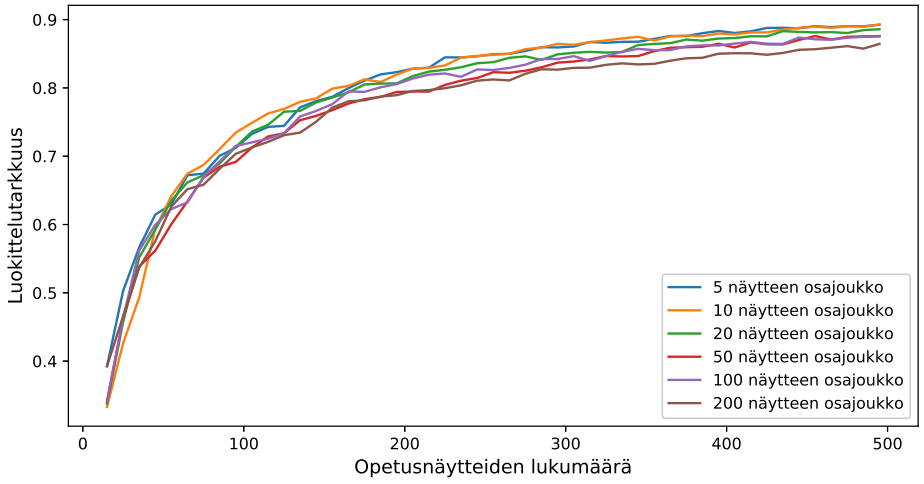
Aktiivisen oppimisen kannalta toinen tärkeä parametri on näytejoukkoihin perustuvan näytteistyksen osajoukon koko. Tämän tarkastelemiseksi valitaan satunnaisen näytteistyksen sijaan epäluotettavuuteen perustuva näytteistysstrategia. Kuvassa 23 ilmenee aktiivisesti opetettujen satunnaismetsä-luokittelijoiden oppiminen erikokoisilla osajoukoilla. Tulos on esitetty keskiarvotettuna viidestä oppimiskäyrästä.

Alle 50 opetusnäytteellä erot ovat pieniä. Opetusdatan kasvaessa oppimiskäyrien kehittymisestä nähdään selvästi trendi, että mitä pienempi on osajoukon koko, sen parempi on luokittelutarkkuus MNIST-datalle. Tämä ilmiö selittynee sillä,



Kuva 22. Eri luokittelijoiden oppiminen satunnaisella näytteistyksellä MNIST-datalle.

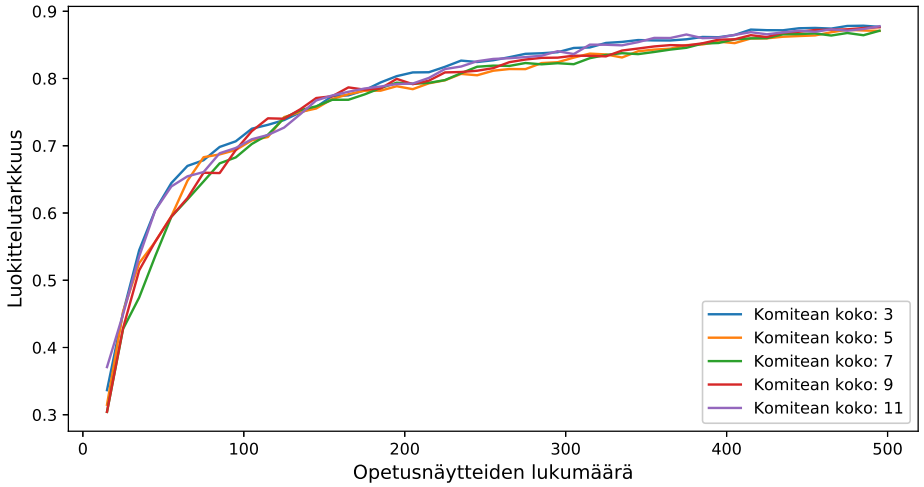
että erityisesti opetuksen alkuvaiheessa on tärkeää, että luokittelijalla on tiedossa riittävä määrä luokkien tyypillisimpiä edustajia. Jos luokittelijalle syötetään alkuun liian monimutkaisia näytteitä, saattaa mallille muodostua väärä kuva luokista ja oppiminen on hidasta. Kuitenkin myöhemmässä vaiheessa oppimista osajoukon koon kasvattamisella on positiivinen vaikutus oppiseen, kun helpot näytteet eivät enää tuo lisäinformaatiota. Paras luokittelutarkkuus kuvan 23 oppimiskäyrien perusteella saatiin 10 näytteen osajoukolla, jota käytetään MNIST-datalle osajoukon kokona.



Kuva 23. Satunnaismetsän oppimiskäyrät epäluotettavuuteen perustuvalla näytteistyksellä erikokoisilla osajoukoilla MNIST-datalle.

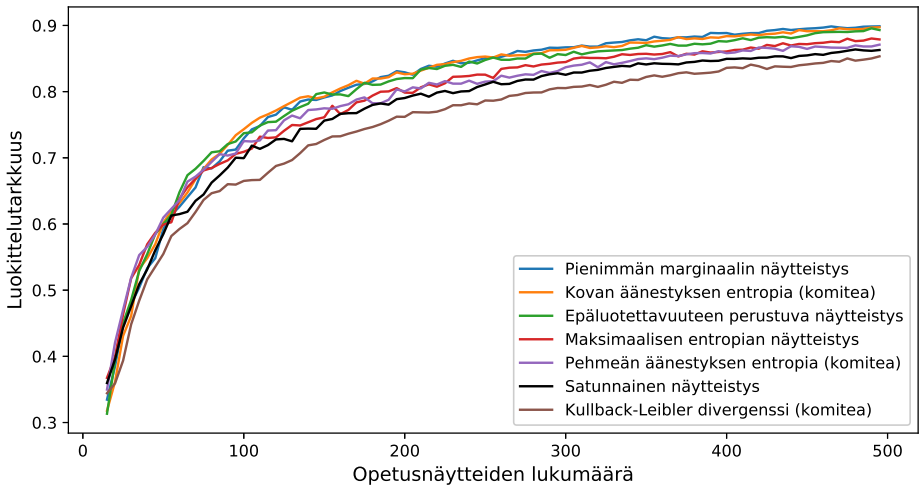
Komiteakyselyssä tulee selvittää, kuinka paljon komitean mallien lukumäärällä on vaikutusta MNIST-datan luokittelutuloksiin. Komitean koon vaikutus oppimiseen havaitaan kuvasta 24, jossa pehmeän äänestyksen entropialla komitean koon kasvattamisella ei ole merkittävää vaikutusta oppimiseen. Koska komitean mallien

lukumäärä on suoraan verrannollinen laskenta-aikaan, käytetään komiteassa kolmea mallia lopullisessa näytteistysstrategioiden vertailussa MNIST-datalle.



Kuva 24. Satunnaismetsän oppimiskäyrät pehmeän äänestyksen entropialla erikokoisilla komiteoilla MNIST-datalle.

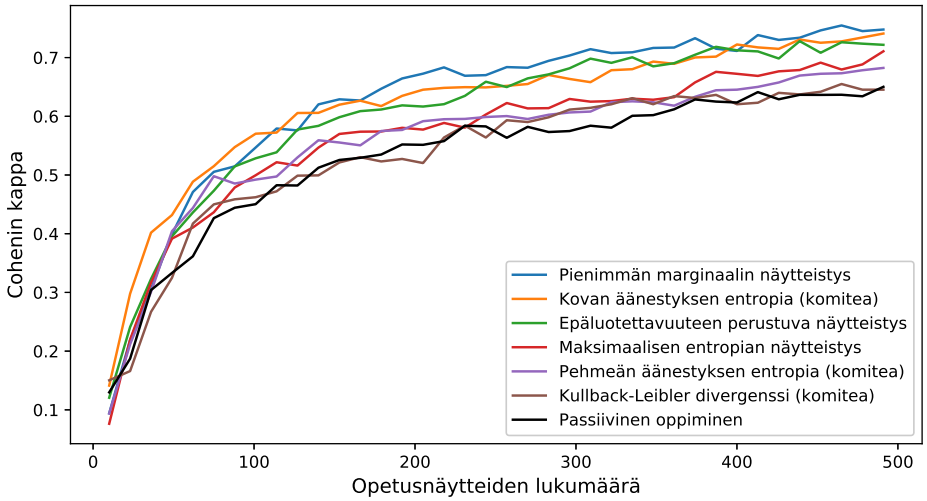
Lopuksi vertaillaan, mikä näytteistysstrategioista toimii parhaiten MNIST-datalle selvitetyillä oppimisparametreilla. Kuvassa 25 on oppimiskäyrät MNIST-datalle eri näytteistysstrategioilla. Näytteistysstrategiana Kullback-Leibler divergenssi toimi huonommin kuin satunnainen näytteistys. Kovan äänestyksen entropialla saavutettiin 80 % luokittelutarkkuus 156 näytteellä, kun satunnaisella näytteistyksellä samaan tarkkuuteen tarvittiin 233 näytettä.



Kuva 25. Satunnaismetsän eri näytteistysstrategioiden oppimiskäyrät MNIST-datalle luokittelutarkkuudella mitattuna.

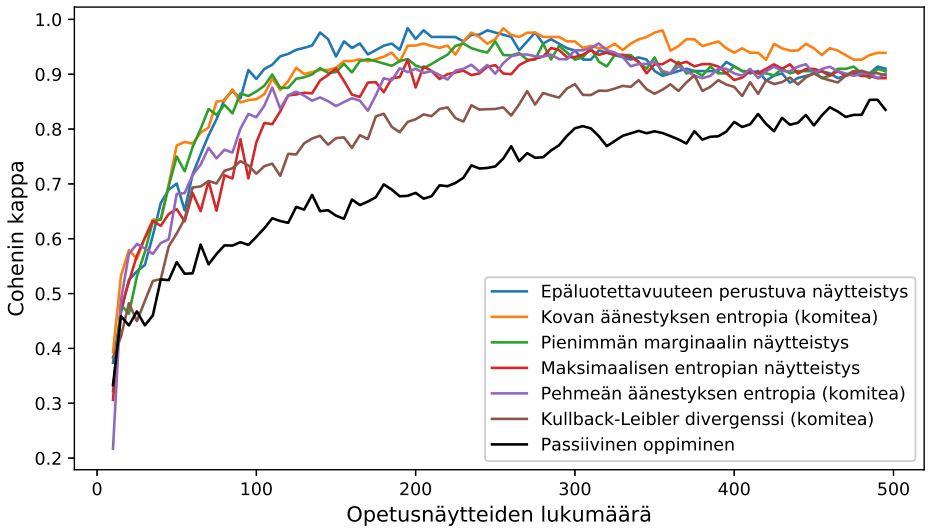
5.4. Perinteinen aktiivinen oppiminen oksadatalle ja madonmunadatalle

Samalla tavalla kuin MNIST-näytteistölle etsitään oksa- ja madonmunadatalle parhaat aktiivisen oppimisen parametrien arvot liitteen 1 kuvaajia tulkitsemalla. Molemmat materiaalit ovat luokkaepäbalanssissa, minkä takia käytetään Cohenin kappakerrointa. Kuvasta 26 ilmenee oksadatalla opetetun satunnaismetsän Cohenin kappakertoimen kehittyminen eri näytteistysstrategioilla, kun osajoukon koko on 20 ja komiteakyselyn näytteistysstrategioissa on kolme mallia. Kuvaajasta nähdään, että mikään näytteistysstrategioista ei ole koko aktiivisen oppimisprosessin ajan selkeästi paras. Cohenin kappakerroin 60 % saavutettiin kovan äänestyksen entropialla 127 näytteellä, kun satunnaisella näytteistyksellä saman Cohenin kappakertoimen saavuttamiseksi tarvittiin 335 näytettä.



Kuva 26. Satunnaismetsän eri näytteistysstrategioiden oppimiskäyrät oksadatalle Cohenin kappakertoimella mitattuna.

Kuvasta 27 ilmenee madonmunadatalla opetetun satunnaismetsän Cohenin kappakertoimen kehittyminen eri näytteistysstategioilla, kun osajoukon koko on 100 ja komiteakyselyn näytteistysstrategioissa on kolme mallia. Cohenin kappakerroin 90 % saavutettiin epäluotettavuuteen perustuvalla näytteistyksellä 95 näytteellä, kun satunnaisella näytteistyksellä jo 80 % Cohenin kappakertoimen saavuttamiseksi vaadittiin 300 näytettä.



Kuva 27. Satunnaismetsän eri näytteistysstrategioiden oppimiskäyrät madonmunadatalle Cohenin kappakertoimella mitattuna.

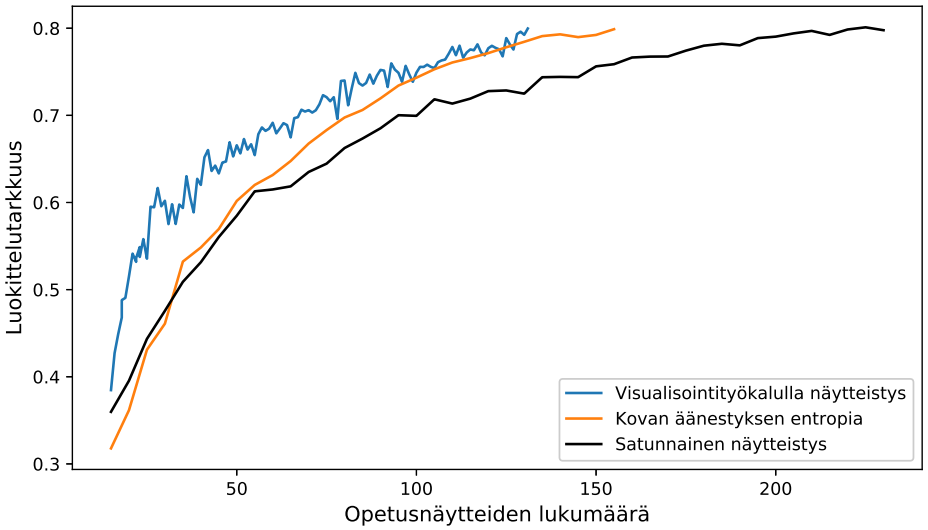
5.5. Aktiivinen oppiminen datan visualisointia apuna käyttäen

Datan visualisoinnista aktiivisen oppimisen aikana pyritään hyötymään näkemällä luokitteluongelman luokkarajojen läheisyyteen. Vaikka aktiivisen oppimisen näytteistysstrategioilla onnistutaan löytämään mallille kaikista vaikeimmin luokiteltavia näytteitä, ei niitä kannata lisätä suoraan opetusdataan tarkastamatta, miltä ne näyttävät. Jos näyte ei muistuta mitään luokkaa, johtaa se opetettavaa mallia harhaan. Datan interaktiivinen visualisoiminen näyttää myös millaisia näytteitä eri luokkiin on jo ennestään kategorisoitu. Näin asiantuntijan on helppo päättää, millaisia luokkien edustajia mallin on syytä oppia tuntemaan paremmin.

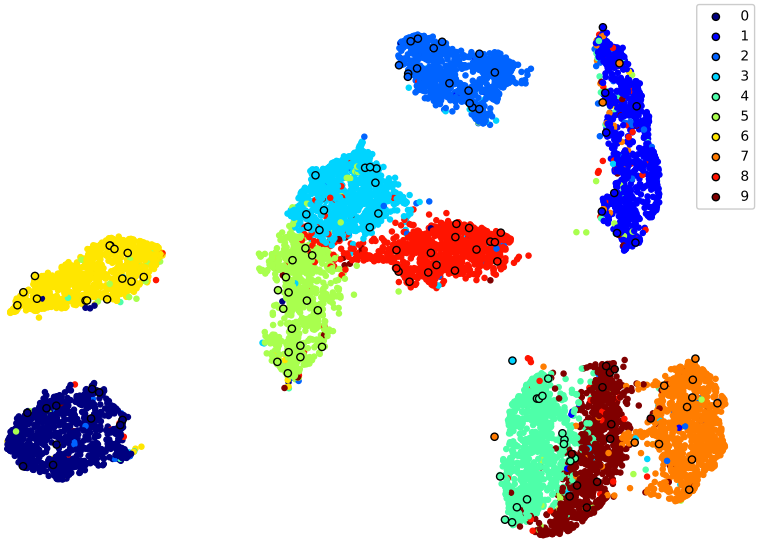
Yleisesti ottaen kaikille datoille visualisointityökalulla kategorisoitaessa paras tulos saavutettiin kategorisoimalla näytteitä, jotka sijoittuivat väärän luokan läheisyyteen, mutta eivät olleet poikkeavia havaintoja (engl. outlier). Lisäksi ilmeni, että jos malli oli epävarma jostain luokasta, suurin osa visualisointityökalulla esitetyistä kategorisoimattomista näytteistä sijoittui tämän luokan läheisyyteen kaksiulotteisessa visualisoinnissa.

MNIST-dataa kategorisoitaessa visualisointityökalulla mallina käytettiin satunnaismetsää. Tällöin osajoukon koko oli 50 ja näytteistysstrategia kovan äänestyksen entropia kolmella komitean jäsenellä. Kuvassa 28 on visualisointityökalulla näytteistyksen oppimiskäyrä MNIST-datalle, jonka lisäksi kuvaajasta löytyvät oppimiskäyrät parhaalla näytteistysstrategialla ja satunnaisella näytteistyksellä. MNIST-datan kohdalla visualisointityökalulla näytteistettäessä saavutettiin 80 % luokittelutarkkuus hieman vähemmällä määrällä opetusdataa kovan äänestyksen entropiaan verrattuna. Käyristä huomataan myös, että erityisesti aktiivisen oppimisprosessin alkuvaiheessa oppiminen on nopeinta visualisointityökalua käytettäessä. Kuvassa 29 on visualisoitu UMAP-algoritmilla visualisointityökalulla kategorisoidut MNIST-datanäytteet kategorisoimattomia

näytteitä vasten. Kategorisoidut mustareunaiset näytteet on valittu reunattomista kategorisoimattomista näytteistä. Toisiaan lähimpänä kaksiulotteisessa kuvauksessa ovat käsinkirjoitetut numerot 3, 8 ja 5 sekä numerot 4, 9 ja 7. Visualisoinnista nähdään, että valituista näytteistä suurin osa on luokkarajojen lähellä ja valitut näytteet kattavat koko data-avaruuden.

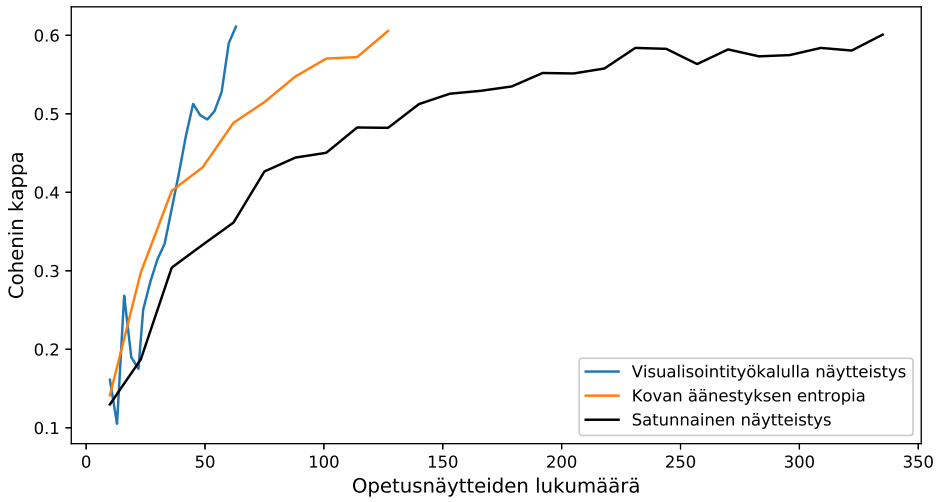


Kuva 28. Satunnaismetsän oppimiskäyrä MNIST-datalle visualisointityökalulla näytteistettäessä verrattuna kovan äänestyksen entropiaan ja satunnaiseen näytteistykseen.



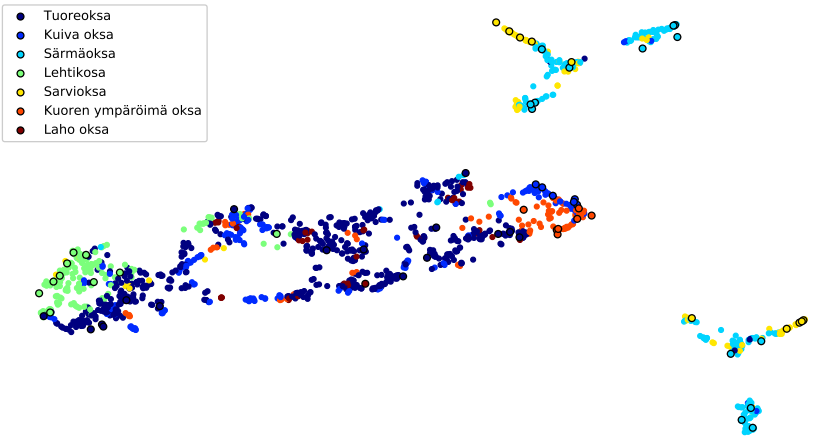
Kuva 29. MNIST-datan visualisointityökalulla kategorisoitujen näytteiden visualisoiminen kaksiulotteisessa data-avaruudessa.

Oksadatan kategorisoimisessa mallina käytettiin satunnaismetsää, osajoukon koko oli 30, näytteistysstrategiana kovan äänestyksen entropia kolmella komiteajäsenellä. Kuvassa 30 on visualisointityökalulla tuotettu oppimiskäyrä oksadatalle, oppimiskäyrä kovan äänestyksen entropialla ja satunnaisella näytteistyksellä. Visualisointityökalulla näytteistettäessä päästiin selvästi vähemmällä määrällä opetusnäytteitä 60 % Cohenin kappakertoimeen kovan äänestyksen entropiaan verrattuna. Alussa oksadatan oppiminen visualisointityökalulla oli hidasta, mutta oppimisprosessin edetessä se nopeutui merkittävästi. Kuvassa 31 on visualisoitu UMAP-algoritmilla visualisointityökalulla kategorisoidut oksanäytteet kategorisoimattomia näytteitä vasten. Nyt kategorisoitavien näytteiden sijainnista on vaikeampi muodostaa johtopäätöstä, kun visualisoinnin luokkien rakenteisuus ei ole selkeä. Kaksiulotteisessa kuvauksessa luokka ”tuoreoksa” sekoittuu luokan ”kuiva oksa” kanssa ja luokka ”särmäoksa” sekoittuu luokan ”sarvioksa” kanssa. Selkeitä klustereita on havaittavissa ainoastaan luokille ”lehtioksa” ja ”kuoren ympäröivä oksa”. Pääsääntöisesti kategorisoidut näytteet ovat kuitenkin lähempänä klustereiden reunoja kuin niiden keskiosaa ja näytteistys on tapahtunut tasaisesti ympäri data-avaruutta.



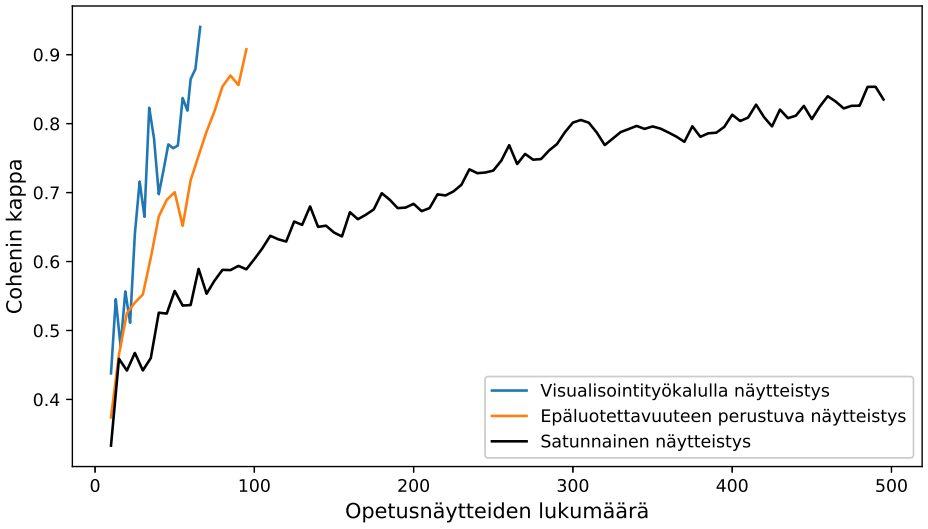
Kuva 30. Satunnaismetsän oppimiskäyrä oksadatalle visualisointityökalulla näytteistettäessä verrattuna kovan äänestyksen entropiaan ja satunnaiseen näytteistykseen.

Madonmunadatan kategorisoimisessa mallina käytettiin satunnaismetsää, osajoukon koko oli 30 ja näytteistysstrategia epäluotettavuuteen perustuva näytteistys. Kuvassa 32 on visualisointityökalulla saatu oppimiskäyrä madonmunadatalle, oppimiskäyrä epäluotettavuuteen perustuvalla näytteistyksellä ja satunnaisella näytteistyksellä. Visualisointityökalulla näytteistettäessä päästiin vähemmällä määrällä opetusdataa 90 % Cohenin kappakertoimeen epäluotettavuuteen perustuvaan näytteistykseen verrattuna. Kuvassa 33 on visualisoitu UMAP-algoritmilla visualisointityökalulla kategorisoidut madonmunanäytteet kategorisoimattomia näytteitä vasten. Visualisoinnista nähdään, että luokka ”heisimato” sekoittuu helposti luokan ”kokkidioosi” kanssa ja luokka ”strongylidi sekoittuu luokan ”nematodirus battus” kanssa. Lisäksi luokkien ”heisimato” ja ”kokkidioosi” välisen luokkarajan

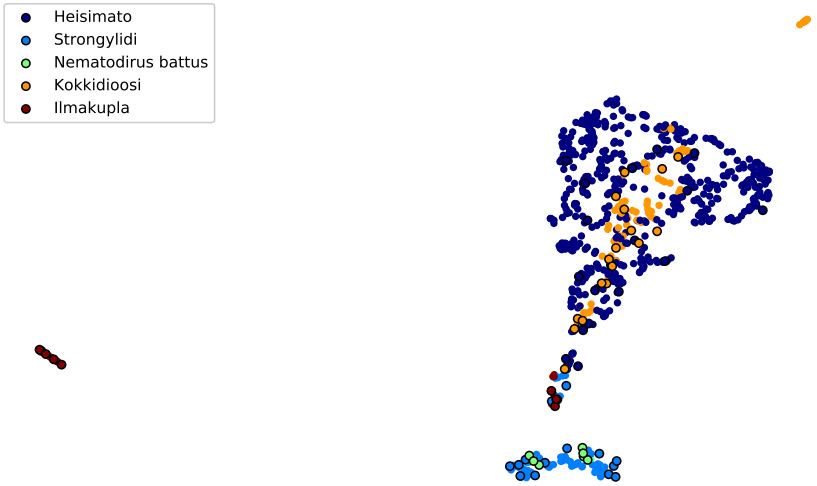


Kuva 31. Oksadatan visualisointityökalulla kategorisoitujen näytteiden visualisoiminen kaksiulotteisessa data-avaruudessa.

kohdalla huomataan selkeästi, kuinka kategorisoidut näytteet sijoittuvat aivan luokkarajojen reunoille. Näytteitä on valittu tasaisesti koko data-avaruuden luokista pois lukien luokan ”kokkidioosi” oikean yläkulman klusteri.



Kuva 32. Satunnaismetsän oppimiskäyrä madonmunadatalle visualisointityökalulla näytteistettäessä verrattuna epäluotettavuuteen perustuvaan näytteistykseen ja satunnaiseen näytteistykseen.



Kuva 33. Madonmunadatan visualisointityökalulla kategorisoitujen näytteiden visualisoiminen kaksiulotteisessa data-avaruudessa.

5.6. Kategorisointityön vähentäminen

Asiantuntijan kategorisointityötä voidaan vähentää entisestään oppimalla datalle tehokas esitystapa ennen aktiivisen oppimisen käyttämistä. Näin tehtiin lyijykynän aihioiden luokittelun oppimiseksi mahdollisimman vähällä määrällä opetusnäytteitä [37]. Siinä 1000 kategorisoimattomalla näytteellä opetettiin konvoluutioautoenkooderi tehokkaan esitystavan löytämiseksi. Tämän jälkeen komiteakyselyssä satunnaismetsä arvioi epävarmuuteen perustuvalla näytteistyksellä kategorisoitavaksi syötettävät näytteet. UMAP-algoritmilla suoritettavat visualisoinnit osoittavat, kuinka konvoluutioautoenkooderin tuottamat piirteet visualisoituvat selvästi erilleen kaksiulotteisessa piirreavaruudessa.

6. POHDINTA

Diplomityön tulokset osoittavat, että datan visualisoiminen nopeuttaa mallin oppimista. Oppimisen nopeutuminen näkyy erityisesti oppimisprosessin alkuvaiheessa, kun mallin tieto luokista on vähäistä. Silloin perinteisen aktiivisen oppimisen menetelmät saattavat tarjota kategorisoitavaksi tehokkaan oppimisen kannalta vääriä näytteitä tai poikkeavia havaintoja. Tämänkaltaiset näytteet asiantuntija huomaa helposti visualisointityökalulla ja jättää lisäämättä ne opetusdataan.

Aktiivisen oppimisen oppimisstrategiaksi valittiin näytejoukkoihin perustuva näytteistys, sillä se valitsee osajoukosta jokaiselle syklille vakiomäärän kategorisoimattomia näytteitä. Tällöin asiantuntijalla on parempi mahdollisuus vaikuttaa mallin oppimiseen kuin esimerkiksi näytteiden virtaukseen perustuvassa näytteistyksessä, jossa asiantuntijan on kategorisoitava tarjottu datanäyte.

Nopean oppimisen kannalta toinen tärkeä ominaisuus on opetusdatan balanssi, joka on visualisointityökalulla kontrolloitavissa. Tämä ei ole mahdollista perinteisillä aktiivisen oppimisen näytteistysstaregioilla, sillä mallilla ei ole kategorisoimattomille näytteille etukäteen tietoa niiden luokista. Parhaiden luokittelutuloksien saavuttaminen vaatii poikkeuksetta balansoidun opetusdatan.

Datan visualisoimiseksi on käytettävissä joukko dimensionaalisuuden vähentämismenetelmiä, jotka huomioivat datan eri ominaisuuksia. Korkeadimensioisen datan muunnoksessa kaksidimensioiseksi informaatiota menee väistämättä hukkaan. Tärkeää on, että käytettävä menetelmä onnistuu säilyttämään korkeadimensioisen datan rakenteen matalaulotteisessa avaruudessa. Tämän ohella menetelmän tulee kyetä säilyttämään datan epälineaarisia riippuvuussuhteita. Nämä ehdot huomioon ottaen parhaimpia menetelmiä datan visuaalisen kuvauksen luomiseksi ovat t-SNE ja UMAP. Jos datassa on monta muuttujaa tai se sisältää monia näytteitä, kestää t-SNE-algoritmilla kauan muunnoksen suorittamisessa. Tästä syystä aktiivisen oppimisen näkökulmasta UMAP-algoritmi on paras menetelmä datan visualisoinnissa.

Tuloksissa löydettiin ominaisuuksiltaan kolmelle erilaiselle datalle vastaukset tutkimuksen alussa esitettyihin tutkimuskysymyksiin. Kaikille datoille minimimäärä näytteitä tietyn luokittelutuloksen saavuttamiseksi datan visualisointityökalulla näytteistettäessä on pienempi kuin parhaalla aktiivisen oppimisen näytteistysstrategialla. Minimimäärään näytteitä vaikuttaa kategorisointityötä tekevän ihmisen asiantuntemus luokitteluongelmaan. MNIST-datan kategorisoiminen ei vaadi niin suurta erityisosaamista kuin oksa- ja madonmunadatan kategorisoiminen.

Myös sattumalla on vaikutusta oppimiseen luokittelijaa opetettaessa, kun osajoukkoon arvottavien näytteiden sopivuus opetusdataksi vaihtelee ajokertojen välillä. Sattuman vaikutus on minimoitu perinteisen aktiivisen oppimisen tuloksista keskiarvottamalla oppimiskäyrät. Sattuman vaikutuksesta huolimatta datan visuaalinen kuvaus tuo helpotusta asiantuntijan kategorisoimistyöhön ja oikeat valinnat johtavat selkeästi nopeampaan oppimiseen.

Lopuksi pystyttiin osoittamaan, että datan visualisointityökalulla kategorisoidut näytteet sijoittuvat pääasiassa data-avaruuden luokkarajojen läheisyyteen. MNIST-

datalle ja madonmunadatalle tämän osoittaminen oli yksiselitteisempää toisin kuin oksadatalle, jonka luokkien klusterit sekoittuivat selvemmin.

Kategorisoimistyötä voidaan vähentää entisestään, kun datalle opitaan tehokas esitystapa. Tällöin opitussa piirreavaruudessa näytteet ovat helpommin erotettavissa omiin luokkiinsa kuin alkuperäisessä data-avaruudessa. Tehokkaan esitystavan oppiminen vaatii suuren määrän näytteitä sekä oikean menetelmän piirteiden irrottamiseksi. Yllättävää oli, että HOG-piirteiden ja LBP-piirteiden käyttäminen tutkimuksessa käytetyille datoille ei tuonut merkittävää parannusta luokittelutuloksiin. Toisaalta tutkimuksen pääpaino ei ollut tehokkaimpien piirteiden löytämisessä datoille. Tärkeämpää oli osoittaa datan visualisoimisen hyöty näytteitä kategorisoidessa suhteessa perinteisiin aktiivisen oppimisen menetelmiin.

Sovellukseen luotavaa luokittelijaa varten tulisi jatkossa selvittää datan tehokkaiden esitystapojen vaikutus kategorisoimisprosessiin, kun apuna on käytetty visualisointeja. Toisena jatkotutkimuskohteena olisi mielenkiintoista tutkia, voisiko datan visualisoinnilla paljastaa, mitkä piirteet sopivat parhaiten eri datoille visualisoitujen klustereiden erottuvuuden perusteella?

7. YHTEENVETO

Tässä diplomityössä osoitettiin, että datan visualisointi helpottaa ja nopeuttaa ihmisen kategorisoimistyötä. Tutkimuksessa lähdettiin liikkeelle esittelemällä eri aktiivisen oppimisen suuntauksia ja näytteistys-strategioita. Tämän jälkeen käytiin läpi eri dimensionaalisuuden vähentämismenetelmiä kaksiulotteisen kuvauksen toteuttamiseksi. Datan visualisointityökalulle sopivimmaksi aktiivisen oppimisen menetelmäksi valikoitui näytejoukkoihin perustuva näytteistys ja dimensionaalisuuden vähentämismenetelmäksi UMAP-algoritmi.

Tulosten luotettavuuden lisäämiseksi datan visualisoimisen vaikutus oppimiseen havainnollistettiin MNIST-, oksa- ja madonmunadatalle. Aktiivisen oppimisen mallina satunnaismetsällä saatiin hyviä tuloksia kaikille datoille. Optimaalinen osajoukon koko ja näytteistysstrategia oli datasta riippuvaista. Datan esikäsittelyllä HOG-piirteillä ja LBP-piirteillä ei ollut luokittelutuloksia parantavaa vaikutusta.

Hyviin luokittelutuloksiin päästiin visualisointyökalua käyttämällä huomattavasti vähemmällä määrällä opetusnäytteitä perinteisiin aktiivisen oppimisen menetelmiin verrattuna. Luokittelutarkkuudella 80 % toimivan satunnaismetsäluokittelijan MNIST-datan opetusnäytteiden määrä putosi 156 näytteestä 134 näytteeseen. Oksadatalle 60 % Cohenin kappakertoimella toimivan satunnaismetsän opetusnäytteiden määrää saatiin pudotettua 127 näytteestä 63 näytteeseen. Madonmunadatalle visualisoinneista oli myös hyötyä, kun 90 % Cohenin kappakertoimella toimivan satunnaismetsän opettaminen väheni 95 näytteestä 65 näytteeseen.

Lopuksi visualisointityökalun kategoriavalintojen visualisoimisella havainnollistettiin, että luokkarajojen läheisyydessä sijaitsee tehokkaan oppimisen kannalta informatiivisimmat näytteet. Parhaan tuloksen saavuttamiseksi mallilla tulee olla lisäksi kattava kuva koko data-avaruudesta.

8. LÄHTEET

[1] Deng J., Dong W., Socher R., Li L.J., Li K. & Fei-Fei L. (2009), ImageNet: A Large-Scale Hierarchical Image Database. DOI: http://doi.org/10.1109/CVPR.2009.5206848.

[2] Silvén O., Niskanen M. & Kauppinen H. (2003) Wood inspection with non-supervised clustering. Machine Vision and Applications 13(5-6), 275–285. DOI: https://doi.org/10.1007/s00138-002-0084-z.

[3] Krizhevsky A., Sutskever I. & Hinton G.E. (2012), Imagenet classification with deep convolutional neural networks, 1097-1105. DOI: https://doi.org/10.1145/3065386.

[4] Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A.C. & Fei-Fei L. (2015) Imagenet large scale visual recognition challenge. International journal of computer vision 115(3), 211–252. DOI: http://doi.org/10.1007/s11263-015-0816-y.

[5] Chapelle O., Scholkopf B. & Zien A. (2006) Semi-supervised learning. MIT Press.

[6] Vapnik V. (1998) Statistical learning theory. Wiley.

[7] Belkin M., Niyogi P. & Sindhwani V. (2006) Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of machine learning research 7(Nov), 2399–2434.

[8] Ratle F., Camps-Valls G. & Weston J. (2010) Semisupervised neural networks for efficient hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing 48(5), 2271–2282. DOI: http://doi.org/10.1109/TGRS.2009.2037898.

[9] Rosenberg C., Hebert M. & Schneiderman H. (2005) Semi-Supervised Self-Training of Object Detection Models. WACV/MOTION, 2. DOI: http://doi.org/10.1109/ACVMOT.2005.107.

[10] Settles B. (2012) Active learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

[11] Lughofer E. (2012) Single-pass active learning with conflict and ignorance. Evolving Systems 3(4), 251–271. DOI: http://doi.org/10.1007/s12530-012-9060-7.

[12] Kingma D.P. & Welling M. (2013), Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[13] LeCun Y. (1998), The mnist database of handwritten digits. URL: http://yann.lecun.com/exdb/mnist/. Luettu 1.3.2020.

[14] Scheffer T., Decomain C. & Wrobel S. (2001) Active hidden markov models for information extraction. In International Symposium on Intelligent Data Analysis 27(3), 309–318. DOI: http://doi.org/10.1007/3-540-44816-0_31.

[15] Shannon C.E. (1948) A mathematical theory of communication. Bell system technical journal 27(3), 379–423. DOI: http://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[16] Cortes C. & Vapnik V. (1995) Support-vector networks. Machine learning 20(3), 273–297. DOI: http://doi.org/10.1007/BF00994018.

[17] Chapelle O. (2007) Training a support vector machine in the primal. Neural computation 19(5), 1155–1178. DOI: http://doi.org/10.1162/NECO.2007.19.5.1155.

[18] Hotelling H. (1933) Analysis of a complex of statistical variables into principal components. Journal of educational psychology 6, 417–441. DOI: http://doi.org/10.1037/h0071325.

[19] Duda R.O., Hart P.E. & Stork D.G. (2012) Pattern classification. John Wiley & Sons.

[20] Torgerson W.S. (1952) Multidimensional scaling: I. Theory and method. Psychometrika 17(4), 401–419. DOI: https://doi.org/10.1007/BF02288916.

[21] Sammon Jr. J.W. (1969) A nonlinear mapping for data structure analysis. IEEE Transactions on Computers 18, 401–409. DOI: https://doi.org/10.1109/T-C.1969.222678.

[22] Kruskal J.B. (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29, 1–27. DOI: https://doi.org/10.1007/BF02289565.

[23] Tenenbaum J.B., De Silva V. & Langford J.C. (2000) A global geometric framework for nonlinear dimensionality reduction. science 290(5500), 2319–2323. DOI: https://doi.org/10.1126/science.290.5500.2319.

[24] Roweis S.T. & Saul L.K. (2000) Nonlinear dimensionality reduction by locally linear embedding. science 290(5500), 2323–2326. DOI: https://doi.org/10.1126/science.290.5500.2323.

[25] Kohonen T. (1982) Self-organized formation of topologically correct feature maps. Biological cybernetics 43(1), 59–69. DOI: https://doi.org/10.1007/BF00337288.

[26] Kramer M.A. (1991) Nonlinear principal component analysis using autoassociative neural networks. AIChE journal 37(2), 233–243. DOI: https://doi.org/10.1002/aic.690370209.

[27] Van Der Maaten L. & Hinton G. (2008) Visualizing data using t-SNE. Journal of machine learning research 9(Nov), 2579–2605.

[28] Roth H.R., Lee C.T., Shin H.C., Seff A., Kim L., Yao J., Lu L. & Summers R.M. (2015) Anatomy-specific classification of medical images using deep convolutional nets. In 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI). DOI: http://doi.org/10.1109/ISBI.2015.7163826.

[29] Jung H., Lee S., Yim J., Park S. & Kim J. (2015), Joint fine-tuning in deep neural networks for facial expression recognition. In Proceedings of the IEEE international conference on computer vision, 2983-2991. DOI: http://doi.org/10.1109/ICCV.2015.341.

[30] Van Der Maaten L. (2014) Accelerating t-SNE using tree-based algorithms. The Journal of Machine Learning Research 15(1), 3221–3245.

[31] McInnes L., Healy J. & Melville J. (2018), Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

[32] Ding J., Condon A. & Shah S.P. (2018) Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nature communications 9(1). DOI: https://doi.org/10.1038/s41467-018-04368-5.

[33] Kauppinen H. & Silvén O. (1996) The effect of illumination variations on color-based wood defect classification. In Proceedings of 13th International Conference on Pattern Recognition Vol. 3, 828–832. DOI: https://doi.org/10.1109/ICPR.1996.547284.

[34] Wan L., Zeiler M., Zhang S., Le Cun Y. & Fergus R. (2013) Regularization of neural networks using dropconnect. In International conference on machine learning, 1058-1066.

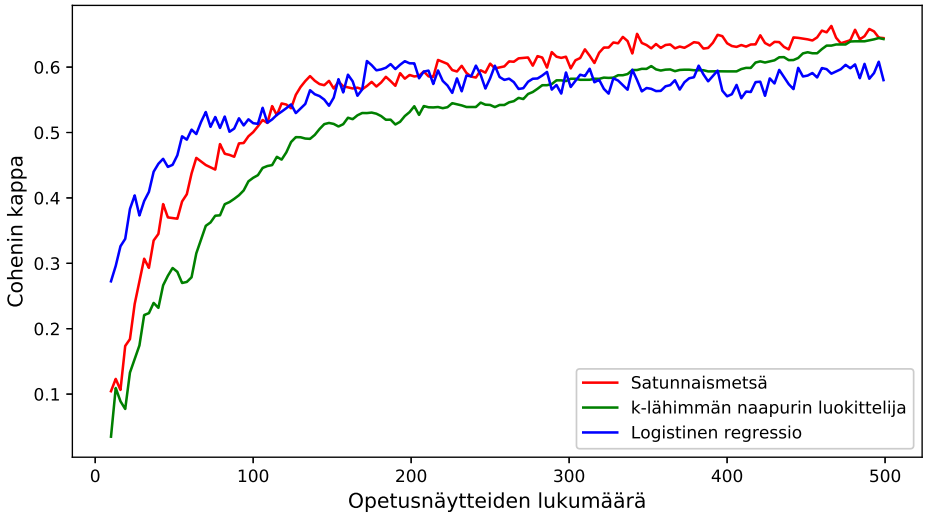
[35] Dalal N. & Triggs B. (2005) Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition Vol. 1, 886–893. DOI: http://doi.org/10.1109/CVPR.2005.177.

[36] Ojala T., Pietikäinen M. & Mäenpää T. (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence 24(7), 971–987. DOI: http://doi.org/10.1109/TPAMI.2002.1017623.

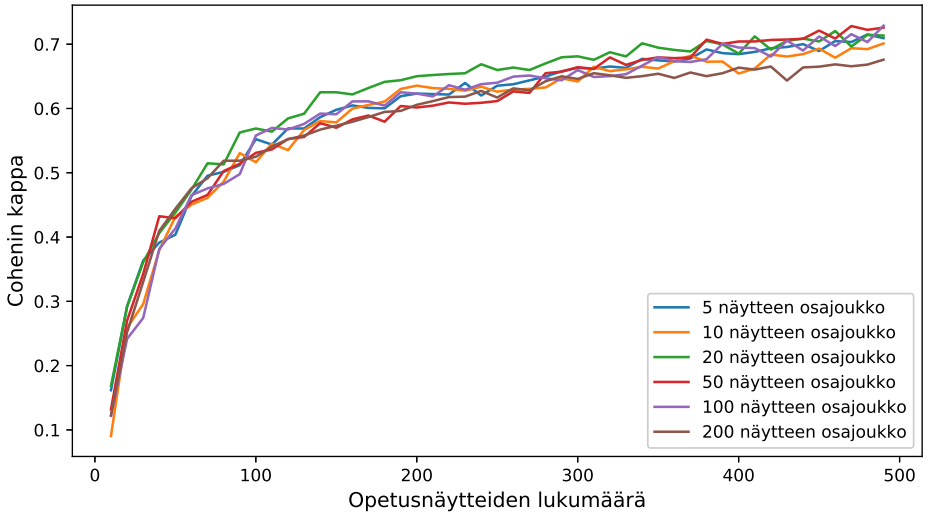
[37] Hosseininia S. (2020), Rapid hybrid learning for visual inspection. URL: https://github.com/SeyyedjavadHosseininia/Rapid-Hybrid-Learning-for-Visual-Inspection. Luettu 10.3.2020.

9. LIITTEET

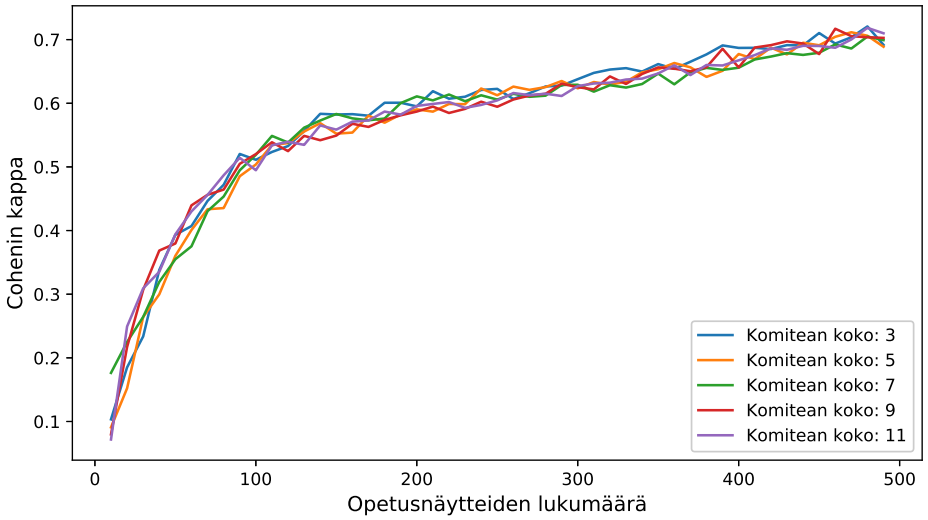
Liite 1. Kuvaajat aktiivisen oppimisen parametrien säätämiseksi oksadatalle ja madonmunadatalle.



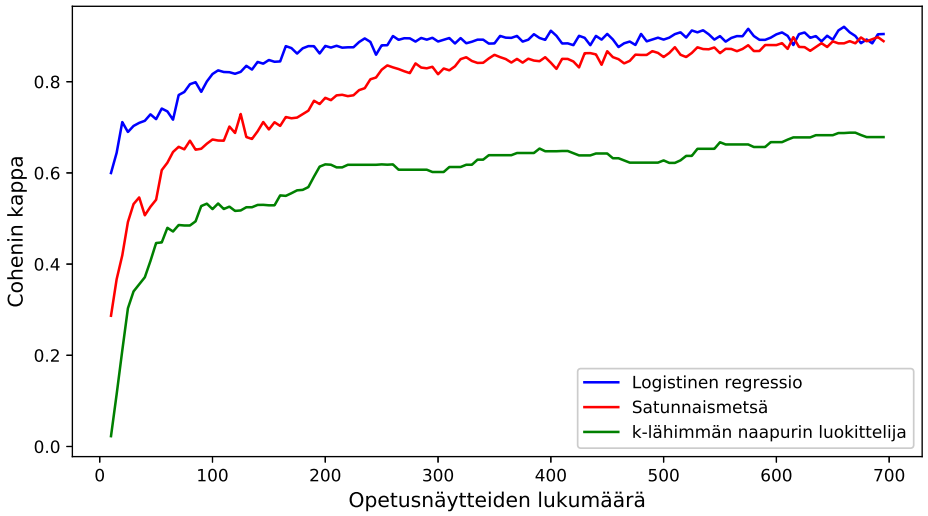
Kuva 1. Eri luokittelijoiden oppimiskäyrät satunnaisella näytteistyksellä oksadatalle Cohenin kappakertoimella mitattuna.



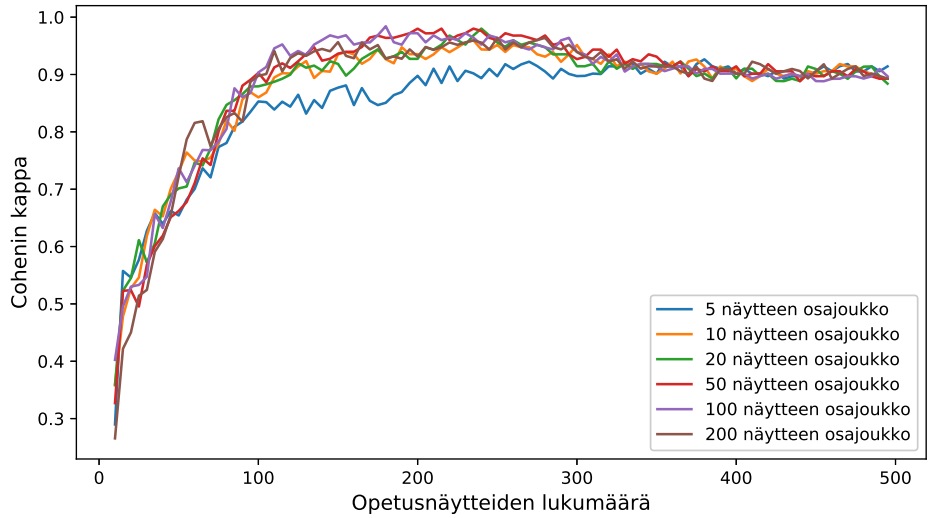
Kuva 2. Satunnaismetsän oppimiskäyrät epäluotettavuuteen perustuvalla näytteistyksellä erikokoisilla osajoukoilla oksadatalle Cohenin kappakertoimella mitattuna.



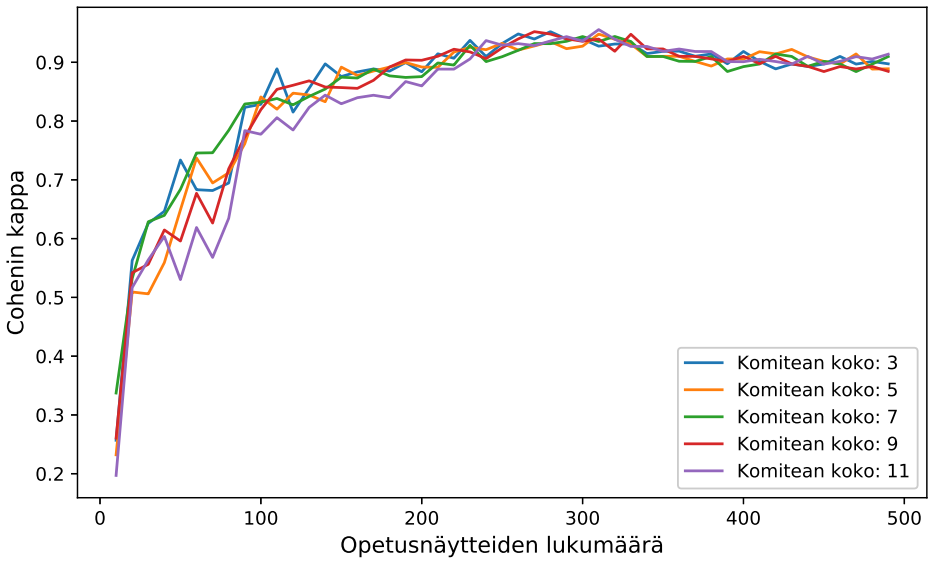
Kuva 3. Satunnaismetsän oppimiskäyrät pehmeän äänestyksen entropialla erikokoisilla komiteoilla oksadatalle Cohenin kappakertoimella mitattuna.



Kuva 4. Eri luokittelijoiden oppimiskäyrät satunnaisella näytteistyksellä madonmunadatalle Cohenin kappakertoimella mitattuna.



Kuva 5. Satunnaismetsän oppimiskäyrät epäluotettavuuteen perustuvalla näytteistyksellä erikokoisilla osajoukoilla madonmunadatalle Cohenin kappakertoimella mitattuna.



Kuva 6. Satunnaismetsän oppimiskäyrät pehmeän äänestyksen entropialla erikokoisilla komiteoilla madonmunadatalle Cohenin kappakertoimella mitattuna.